



St.-Bernhard-Gymnasium
Staatlich genehmigtes privates Gymnasium

 **Malteser**
...weil Nähe zählt.

Schulinternes Curriculum

des St.-Bernhard-Gymnasiums

Informatik

zu den Kernlehrplänen der Erprobungs-, Mittel- und
Gymnasialen Oberstufe

Stand: 26.05.2026 (AC)

Inhaltsverzeichnis

1.	Die Fachgruppe Informatik am St.-Bernhard-Gymnasium Willich	3
2	Entscheidungen zum Unterricht	5
2.1	Unterrichtsvorhaben Erprobungsstufe (6. Klasse)	6
2.2	Unterrichtsvorhaben in der Mittelstufe (Wahlpflichtbereich II)	17
2.3	Unterrichtsvorhaben Sekundarstufe II	26
2.3.1	<i>Übersichtsraster Unterrichtsvorhaben</i>	27
2.3.2	<i>Konkretisierte Unterrichtsvorhaben</i>	41
2.4	Grundsätze der fachmethodischen und fachdidaktischen Arbeit	147
2.5	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	148
2.5.1	<i>Sekundarstufe I</i>	148
2.5.2	<i>Sekundarstufe II</i>	154
3	Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	159
4	Qualitätssicherung und Evaluation	159

1. Die Fachgruppe Informatik am St.-Bernhard-Gymnasium

Seit über 35 Jahren wird am St.-Bernhard-Gymnasium das Fach Informatik unterrichtet. Somit liegen weitreichende Erfahrungswerte im Informatikunterricht vor. Generelle Zielsetzung ist die Aufrechterhaltung eines Informatikunterrichtes von der Erprobungsstufe bis zum Abschluss der Qualifikationsphase. Er ist darauf angelegt, einen sinnvollen und verantwortungsbewussten Umgang mit digitalen Medien und informatischen Konzepten aufzuzeigen und Mädchen und Jungen gleichermaßen die zentralen Ideen der Informatik näherbringen.

Der Informatikunterricht findet insgesamt in vier Computerräumen statt. Diese enthalten jeweils 30 Rechner, einen Drucker und einen Beamer mit angeschlossener Soundanlage. Die Rechner sind in einem Netzwerk integriert, wodurch Videovernetzung, Datenaustausch und andere netzwerkabhängige methodische Instrumente fester Bestandteil des Informatikunterrichts sind. Zusätzlich wird die MS Office Cloud verwendet, um Daten auch über den Computerraum hinaus auszutauschen und zur Verfügung zu stellen. Die Art der Anordnung der Computer begünstigt einen projektorientierten, auf Gruppen- und Partnerarbeit angelegten Unterricht, der sich als zentrale Unterrichtsform etabliert hat. Die Lehrkraft hat in diesem Unterricht mehr moderierende Elemente zu übernehmen und kann stärker auf Schülerinnen und Schüler zugehen. Ebenso ist die Schülerinteraktion innerhalb der Lerngruppe ein wesentliches Element individueller Förderung.

Von der Erprobungsstufe bis hin zur Klasse 8 kommen insbesondere blockorientierte Programmierumgebungen (z. B. Scratch, Blockly, MakeCode) zur Anwendung – sowohl im obligatorischen Informatikunterricht in der Erprobungsstufe, der bei uns in der Jahrgangsstufe 6 zweistündig unterrichtet wird, als auch in verschiedenen Neigungsgruppen und AGs. Wir freuen uns, dass bereits seit mehreren Jahren engagierte Eltern ebenfalls Informatik- und Technik-AGs anbieten.

Im MINT-Kurs der Jahrgangsstufe 8 erfolgt eine spielerische Einführung in die textbasierte Sprache Python. Ebenso können hier die Schüler je nach Interesse weitere Konzepte der Informatik (z. B. Verschlüsselungsverfahren, Arduino etc.) kennenlernen.

Im Differenzierungsbereich der Jahrgangsstufen 9 und 10 wird nach und nach der visuelle Programmieranteil (z. B. Kara der Käfer) zugunsten einer mit Programmieranweisungen

dominierten Sprache (Python) reduziert. Dabei wird darauf geachtet, dass die Aufgabenstellungen und die produzierten Ergebnisse sich leicht über feststehende oder bewegte Bildsequenzen auf die richtige Umsetzung der Algorithmen hin überprüfen lassen. Die weiteren Themen ergeben sich aus dem KLP Informatik für den Differenzierungsbereich (s. u. Unterrichtsvorhaben)

Der Programmierstil wird in diesen beiden Jahrgangsstufen bewusst imperativ umgesetzt. In der SEK II wird dann auf einen objektorientierten Ansatz zur Lösung informatischer Probleme umgeschwenkt (Java). Dabei ist der Abstraktionsgrad der Aufgabenstellungen und der zu entwickelnden Lösungen für schulische Verhältnisse denkbar hoch. Das Ziel ist - neben der Abiturqualifikation - die bestmöglichen Voraussetzungen für ein Studium der Informatik und verwandter Fächer zu ermöglichen.

Bei der Gestaltung der Pläne wird darauf geachtet, dass es für jede Schülerin und jeden Schüler am St.-Bernhard-Gymnasium möglich ist, sowohl im Wahlpflichtbereich II als auch in der Oberstufe komplett neu einsteigen zu können.

Seit einigen Jahren bereits nehmen die SuS an verschiedenen Wettbewerben teil. So ist etwa die Teilnahme am „Informatik-Biber“ sowie an der ersten und ggf. zweiten Runde des JWINF verbindlicher Bestandteil des Fachunterrichts in der 6. Klasse. Darüber hinaus ermutigen die unterrichtenden Lehrer ihre Schüler in den höheren Jahrgangsstufen zur freiwilligen Teilnahme weiterer Wettbewerbe

Neben dem Vermitteln von Sachkenntnissen will die Fachgruppe Informatik unsere Schülerinnen und Schüler dabei unterstützen, Freude am eigenen Nachdenken und eigenem wissenschaftlichen Arbeiten zu entwickeln, sowie ein mündiges Mitglied einer digitalisierten Gesellschaft zu werden. Gemäß den Leitlinien für Erziehung und Bildung an Malteser Gymnasien werden die Schülerinnen und Schüler als einzelne Persönlichkeiten angesehen, deren individuelle Begabungen und Fähigkeiten es zu entdecken und deren Nutzen es zu entfalten gilt. Es gilt auch und gerade für die Informatik, die Chancen und

Gefahren sowie Disruptionen durch Lernende Systeme bzw. künstliche Intelligenzen in den unterschiedlichen Lern- und Lebensbereichen immer stärker in den Blick zu nehmen.

2 Entscheidungen zum Unterricht

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

2.1 Unterrichtsvorhaben Erprobungsstufe (6. Klasse)

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.1: Wir präsentieren uns als Avatar <i>Was ist ein Informatiksystem und wie kann ich es für ein projektartiges Vorhaben nutzen?</i> ca. 6 Ustd.	IF: Information und Daten - Informationsgehalt von Daten IF: Informatiksysteme - Aufbau und Funktionsweise von Informatiksystemen - Anwendung von Informatiksystemen IF: Informatik, Mensch und Gesellschaft - Informatiksysteme in der Lebens- und Arbeitswelt - Datenbewusstsein	Argumentieren (A) - formulieren Fragen zu einfachen informatischen Sachverhalten - äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen - begründen die Auswahl eines Informatiksystems Modellieren und Implementieren (MI) - erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) - beschreiben einfache Darstellungen von informatischen Sachverhalten Kommunizieren und Kooperieren (KK) - beschreiben und erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht - setzen bei der Bearbeitung einer informatischen Problemstellung geeignete digitale Werkzeuge	- stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI) - interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI) - 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 (DI) (MKR 6.1) - vergleichen Möglichkeiten der Datenverwaltung hinsichtlich ihrer spezifischen Charakteristika (u. a. Speicherort, Kapazität, Aspekte der Datensicherheit) (A)

		zum kollaborativen Arbeiten ein (MKR 1.2, 3.1)	• setzen zielgerichtet Informatiksysteme zur Verarbeitung von Daten ein (MI) • erläutern Prinzipien der strukturierten Dateiverwaltung (A) (MKR 1.3) • setzen Informatiksysteme zur Kommunikation und Kooperation ein (KK) (MKR 3.1) • beschreiben an Beispielen die Bedeutung von Informatiksystemen in der Lebens- und Arbeitswelt (KK) (MKR 6.4) • benennen und erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen auf ihre Lebens- und Erfahrungswelt (A/KK) (MKR 6.4) / (VB C Z5) • erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A) (VB C Z3)
Weitere Hinweise, Vereinbarungen und Absprachen: ... zur Vernetzung: Evtl. Rückgriff auf FFG (Jahrgangsstufe 5 → Umgang mit der MS Cloud und den Schul-PCs) • Informatiksysteme werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. in den Unterrichtsvorhaben „Von der Anweisung zum Algorithmus“ (UV 6.2) und „Automaten in unserer Lebenswelt“ (UV 6.4) sowie „Eigene Programme mit dem Calliope Mini“ (UV 6.2)			

... zu Synergien:

- Kunst – ein Portrait wird durch die Reduzierung auf wenige Merkmale abstrahiert
- Eigenverantwortliches Lernen – der erste Zugang zu der schuleigenen Lernplattform und der Umgang mit dieser kommt der Arbeit in allen weiteren Unterrichtsfächern zugute

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.2: Von der Anweisung zum Algorithmus ca. 18 Ustd.	IF: Information und Daten • Daten und ihre Codierung • Informationsgehalt von Daten IF: Algorithmen • Algorithmen und algorithmische Grundkonzepte • Implementation von Algorithmen	Argumentieren (A) • formulieren Fragen zu einfachen informatischen Sachverhalten • äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen Modellieren und Implementieren (MI) • erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) • beschreiben einfache Darstellungen von informatischen Sachverhalten Kommunizieren und Kooperieren (KK) • erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht • setzen bei der Bearbeitung einer informatischen Problemstellung geeignete digitale Werkzeuge	• stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI) • formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI) • überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI) • führen Handlungsvorschriften schrittweise aus (MI) • identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI) (MKR 6.2) • implementieren Algorithmen in einer visuellen Programmiersprache (MI) (MKR 6.1, 6.3)

		zum kollaborativen Arbeiten ein (MKR 1.2, 3.1)	- implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI), (MKR 6.1, 6.3) - überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI) (MKR 6.2)
Weitere Hinweise, Vereinbarungen und Absprachen: - Da einige SuS in der 5. Klasse bereits an der Neigungsgruppe „Scratch“ teilgenommen haben, empfiehlt sich hier eine andere visuelle Programmiersprache, zum Beispiel <i>Blockly (Games)</i> - Teilnahme am Wettbewerb <i>Informatik-Biber</i> ... zur Vernetzung: - Algorithmen werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. in den Unterrichtsvorhaben „<i>Codierungen zum Austausch und zur Verarbeitung von Nachrichten</i>“ (UV 6.3) oder „<i>Automaten in unserer Lebenswelt</i>“ (UV 6.4) ... zu Synergien: - Mathematik – Systematisierung von Rechenoperationen; Umgang mit Tabellenkalkulationsprogrammen			

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.3: Codierungen zum Austausch und zur Verarbeitung von Nachrichten ca. 8 Ustd.	IF: Information und Daten - Daten und ihre Codierung - Informationsgehalt von Daten	Argumentieren (A) - formulieren Fragen zu einfachen informatischen Sachverhalten - äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen Modellieren und Implementieren (MI) - erstellen informatische Modelle zu gegebenen Sachverhalten	- 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)

		Darstellen und Interpretieren (DI) • beschreiben einfache Darstellungen von informatischen Sachverhalten • stellen informatische Sachverhalte in geeigneter Form dar • interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht • erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht • kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme	• nennen Beispiele für die Codierung von Daten aus ihrer Erfahrungswelt (DI) • codieren und decodieren Daten unter Verwendung des Binärsystems (MI) • interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI) • erläutern Einheiten von Datenmengen (A / KK) • <i>vergleichen Datenmengen hinsichtlich ihrer Größe mit Hilfe anschaulicher Beispiele aus ihrer Lebenswelt (DI)</i>
Weitere Hinweise, Vereinbarungen und Absprachen: ... zur Vernetzung: Informationen aus Daten zu entnehmen wird im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. in den Unterrichtsvorhaben „Detektivarbeit – Auseinandersetzung mit verschiedenen Verfahren zur Codierung und Verschlüsselung sowie deren Anwendung“ (UV 6.1), „Künstliche Intelligenz und maschinelles Lernen - Wie können Menschen und Informatiksysteme auf Grundlage von Daten Vorhersagen treffen?“ (UV 6.3) oder „Datenbewusstsein - Welche Informationen kann man aus meinen Daten oder großen Datenmengen über mich ableiten? Was bedeutet dies für mein Datenbewusstsein“ (UV 6.4) ... zu Synergien: • Mathematik – Stellenwertsysteme (Übernahme der Fachbegriffe aus dem Mathematikbuch); Physik: Rechnen mit Einheiten			

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.4: Automaten in unserer Lebenswelt ca. 6 Ustd.	IF: Automaten und künstliche Intelligenz - Aufbau und Wirkungsweise einfacher Automaten IF: Informatiksysteme - Aufbau und Funktionsweise von Informatiksystemen	Argumentieren (A) - formulieren Fragen zu einfachen informatischen Sachverhalten - äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen Modellieren und Implementieren (MI) - erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) - beschreiben einfache Darstellungen von informatischen Sachverhalten Kommunizieren und Kooperieren (KK) - beschreiben und erläutern einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht	- erläutern die Funktionsweise eines Automaten aus ihrer Lebenswelt (A) (MKR 6.1) - stellen Abläufe in Automaten graphisch dar (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 (DI)
Weitere Hinweise, Vereinbarungen und Absprachen: ... zur Vernetzung: Aufbau und Wirkungsweise von Automaten werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Unterrichtsvorhaben „Künstliche Intelligenz und maschinelles Lernen - Wie können Menschen und Informatiksysteme auf Grundlage von Daten Vorhersagen treffen?“ (UV 6.3)			

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.5: Detektivarbeit – Auseinandersetzung mit verschiedenen Verfahren zur Codierung und Verschlüsselung sowie deren Anwendung ca. 6 -7 Ustd.	IF: Information und Daten • Daten und ihre Codierung • Verschlüsselungsverfahren IF: Algorithmen • Algorithmen und algorithmische Grundkonzepte IF: Informatik, Mensch und Gesellschaft • Datensicherheit und Sicherheitsregeln	Argumentieren (A) • formulieren Fragen zu einfachen informatischen Sachverhalten • äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen Modellieren und Implementieren (MI) • erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) • beschreiben einfache Darstellungen von informatischen Sachverhalten • stellen informatische Sachverhalte in geeigneter Form dar • interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht • erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht	• nennen Beispiele für die Codierung von Daten aus ihrer Erfahrungswelt (DI) • codieren und decodieren Daten unter Verwendung des Binärsystems (MI) • erläutern Einheiten von Datenmengen (A/KK) • erläutern ein einfaches Transpositionsverfahren als Möglichkeit der Verschlüsselung (DI) (MKR 1.4) • vergleichen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (DI) (MKR 1.4) • führen Handlungsvorschriften schrittweise aus (MI) • beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A)

		• strukturieren gemeinsam eine Lösung für ein informatisches Problem • dokumentieren gemeinsam ihren Arbeitsprozess und ihre Ergebnisse auch mithilfe digitaler Werkzeuge (MKR 1.2)	
Weitere Hinweise, Vereinbarungen und Absprachen: ... zur Vernetzung: • Informationen aus Daten zu erhalten und diese zu entschlüsseln, spielt im Unterrichtsverlauf in Informatik immer wieder eine Rolle, z. B. in dem Unterrichtsvorhaben „Codierungen zum Austausch und zur Verarbeitung von Nachrichten“ (UV 6.3) ... zu Synergien: • z. B.: Zusammenarbeit mit dem Fach Deutsch, Lesen des Buches Kalle Blomquist (Räubersprache); Geschichte (Cäsar), Mathematik			

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.6: Künstliche Intelligenz und maschinelles Lernen <i>Wie können Menschen und Informatiksysteme auf Grundlage von Daten Vorhersagen treffen?</i> ca. 9 Ustd.	IF: Automaten und künstliche Intelligenz • Maschinelles Lernen mit Entscheidungsbäumen • <i>Maschinelles Lernen mit neuronalen Netzen</i>	Argumentieren (A) • 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 Modellieren und Implementieren (MI)	• 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>

		• erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) • beschreiben einfache Darstellungen von informatischen Sachverhalten • stellen informatische Sachverhalte in geeigneter Form dar • interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht • <i>anstelle der vorherigen KE: erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht</i>	
Weitere Hinweise, Vereinbarungen und Absprachen: ... zur Vernetzung: • Vorhersagungen auf Grundlage von Daten spielen im Unterrichtsverlauf in Informatik immer wieder eine Rolle, z. B. in dem Unterrichtsvorhaben „Detektivarbeit – Auseinandersetzung mit verschiedenen Verfahren zur Codierung und Verschlüsselung sowie deren Anwendung“ (UV 6.1) ... zu Synergien: • ggf. Zusammenarbeit mit dem Fach Biologie (→ Neuronen, Synapsen)			

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Schülerinnen und Schüler</i>	Konkretisierte Kompetenzerwartungen <i>Schülerinnen und Schüler</i>
UV 6.7: Datenbewusstsein <i>Welche Informationen kann man aus meinen Daten oder großen Datenmengen über mich ableiten? Was bedeutet dies für mein Datenbewusstsein?</i> ca. 5 Ustd.	IF: Informatik, Mensch und Gesellschaft • Informatiksysteme in der Lebens- und Arbeitswelt • Datenbewusstsein • Datensicherheit und Sicherheitsregeln	Argumentieren (A) • 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 Darstellen und Interpretieren (DI) • beschreiben einfache Darstellungen von informatischen Sachverhalten • stellen informatische Sachverhalte in geeigneter Form dar • interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht • <i>anstelle der vorherigen KE: erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht</i>	• benennen an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen auf ihre Lebens- und Erfahrungswelt (A/KK) (MKR 6.4) / (VB C Z5) • <i>anstelle der vorherigen KE: erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK), (MKR 6.4) / (VB C Z5)</i> • beschreiben anhand von ausgewählten Beispielen die Verarbeitung und Nutzung personenbezogener Daten (DI) (VB C Z5) • erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A) (VB C Z3) • beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A), (MKR 1.4) / (VB C Z2)

		• kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme	
Weitere Hinweise, Vereinbarungen und Absprachen: ... zur Vernetzung: • Informationen aus Datenmengen abzuleiten und ein Bewusstsein für Daten zu entwickeln, spielt im Unterrichtsverlauf in Informatik immer wieder eine Rolle, z. B. in den Unterrichtsvorhaben „<i>Codierungen zum Austausch und zur Verarbeitung von Nachrichten</i>“ (UV 6.3) und „<i>Detektivarbeit – Auseinandersetzung mit verschiedenen Verfahren zur Codierung und Verschlüsselung sowie deren Anwendung</i>“ (UV 6.1) ... zu Synergien: • Anknüpfungspunkte mit Blick auf Wirtschaft/Politik			

2.2 Unterrichtsvorhaben in der Mittelstufe (Wahlpflichtbereich II)

Jahrgangsstufe 9

Unterrichtsvorhaben I: Überall Automaten – Vom Lichtschalter zum Marienkäfer

Schwerpunkte der Kompetenzentwicklung – Übergeordnete Kompetenzerwartungen:

Die Schülerinnen und Schüler

- strukturieren informatische Sachverhalte (MI),
- entwickeln informatische Modelle zu gegebenen Problemstellungen (MI),
- wenden ein informatisches Verfahren zur Lösung eines Problems an (MI),
- interpretieren unterschiedliche Darstellungen von informatischen Sachverhalten (DI).

Inhaltsfelder: Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

- ♦ Aufbau und Wirkungsweise von Automaten

Konkretisierte Kompetenzerwartungen:

Die Schülerinnen und Schüler

- analysieren die Funktionsweise eines Automaten mit Hilfe eines Zustandsübergangsdiagramms (DI), (MKR 6.3)
- entwickeln einen Automaten für eine konkrete Problemstellung (MI). (MKR 6.3)

Zeitbedarf: ca. 22 Ustd.

Ggf. Absprachen zur Leistungsüberprüfung: /

Verbindliche Hinweise und Absprachen zu diesem Unterrichtsvorhaben / Umsetzung: Anhand des endlichen Automaten „Lichtschalter“ werden die Begriffe „Zustand“, „Übergang“, „Sensor“ und „Aktion“, sowie die grafische Darstellung eines Automaten als Zustandsübergangsdiagramm eingeführt. Diese Begrifflichkeiten werden anschließend auf die Elemente in der Programmierumgebung „Kara“ übertragen. Im Rahmen der Programmierumgebung kann der Marienkäfer „Kara“ verschiedene Aufgaben lösen. Dabei nimmt die Komplexität der Aufgaben immer weiter zu. Die verschiedenen Schwierigkeitsgrade der Aufgaben erlauben besonders gut ein binnendifferenziertes Arbeiten.

Entscheidungen zu fach- und/oder fächerübergreifenden Fragen: /

Schwerpunkte der Kompetenzentwicklung – Übergeordnete Kompetenzerwartungen:

Die Schülerinnen und Schüler

- bewerten informatische Sachverhalte kriteriengeleitet (A),
- strukturieren informatische Sachverhalte (MI),
- analysieren Modelle und Implementierungen (MI),
- entwickeln informatische Modelle zu gegebenen Problemstellungen (MI),
- implementieren informatische Modelle (MI),
- beurteilen Modelle und Implementierungen hinsichtlich der Lösung einer Problemstellung (MI),
- wenden ein informatisches Verfahren zur Lösung eines Problems an (MI),
- identifizieren informatische Sachverhalte in komplexen Anwendungsbereichen (DI),
- veranschaulichen informatische Sachverhalte (DI),
- interpretieren Ergebnisse von Implementierungen (DI),
- interpretieren unterschiedliche Darstellungen von informatischen Sachverhalten (DI),
- stellen informatische Sachverhalte unter Verwendung von Fachbegriffen dar (KK).

Inhaltsfelder: Information und Daten; Algorithmen; Automaten und formale Sprachen; Informatiksysteme

Inhaltliche Schwerpunkte:

- ♦ Erfassung, Verarbeitung und Verwaltung von Daten
- ♦ Algorithmen und algorithmische Grundkonzepte
- ♦ Variablen
- ♦ Implementation von Algorithmen
- ♦ Erstellung und Analyse von Quelltexten
- ♦ Anwendung von Informatiksystemen

Konkretisierte Kompetenzerwartungen:

Die Schülerinnen und Schüler

- verarbeiten Daten mit einer Programmiersprache unter Berücksichtigung logischer und arithmetischer Operationen (MI),
- wählen geeignete Datentypen im Kontext eines Anwendungsbeispiels aus (MI),
- interpretieren Daten aus dem Ergebnis eines Verarbeitungsprozesses (DI),
- überprüfen algorithmische Eigenschaften (Endlichkeit der Beschreibung, Eindeutigkeit, Terminierung) in Handlungsvorschriften (A), (MKR 6.1)
- stellen Algorithmen in verschiedenen Repräsentationen dar (DI), (MKR 6.3)
- entwerfen und implementieren Algorithmen unter Verwendung von Variablen verschiedener Typen und unter Berücksichtigung des Prinzips der Modularisierung (MI), (MKR 6.1, 6.2, 6.3)
- kommentieren, modifizieren und ergänzen Quelltexte von Programmen nach Vorgaben (MI), (MKR 6.3)
- erläutern die Möglichkeit der Werteübergabe mithilfe von Parametern (MI), (MKR 6.1)
- überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen bei der Lösung gleichartiger Probleme (MI), (MKR 6.2, 6.4)
- beurteilen die Problemangemessenheit verwendeter Algorithmen (MI), (MKR 6.4)
- erläutern die Begriffe Syntax und Semantik einer Programmiersprache an Beispielen (KK),

- analysieren Quelltexte auf syntaktische Korrektheit (A/MI), (MKR 6.3)
- erstellen syntaktisch korrekte Quelltexte in einer geeigneten Dokumentenbeschreibungssprache und in einer Programmiersprache (MI), (MKR 6.3)
- wenden zielgerichtet Prinzipien der strukturierten Dateiverwaltung an (MI). (MKR 1.3)

Zeitbedarf: ca. 28 Ustd.

Ggf. Absprachen zur Leistungsüberprüfung: /

Verbindliche Hinweise und Absprachen zu diesem Unterrichtsvorhaben / Umsetzung:

Die Fachkonferenz hat sich auf die textorientierte Programmiersprache Python mit der integrierten IDLE als Entwicklungsumgebung geeinigt. Um den Einstieg in die Verwendung einer textorientierten Programmiersprache zu erleichtern, wird zunächst die von Python zur Verfügung gestellte Turtlegrafik verwendet. Anschließend können andere imperative Python-Programme zu verschiedenen Problemstellungen entworfen und implementiert werden. Programmablaufpläne werden verwendet, um die Funktionsweise von Programmen zu verdeutlichen und Programme oder Methoden zu entwickeln. Die Modularisierung von Algorithmen und Programmen erfolgt durch die Verwendung bzw. Implementation von Methoden. Parameterübergaben werden an verschiedenen Beispielen erläutert. Zu mehreren Problemstellungen wird die Problemangemessenheit der verwendeten Algorithmen beurteilt. Um Werte zu speichern, werden Variablen verschiedener Typen verwendet. Da in Python Variablen nicht deklariert werden müssen, kann die Weiterverarbeitung von Benutzereingaben einen Anlass bieten, Variablentypen zu thematisieren und im Kontext eines Anwendungsbeispiels geeignete Datentypen auszuwählen. Ausgehend von einem nicht terminierenden Programm können einige Handlungsvorschriften und Programmteile auf algorithmische Eigenschaften (Endlichkeit der Beschreibung, Eindeutigkeit, Terminierung) überprüft werden. Zielgerichtetes Testen und die Analyse von Quelltexten auf syntaktische Korrektheit kann sowohl bei der Implementation selbst entwickelter Programmteile als auch im Zusammenhang mit der Überprüfung der Wirkungsweise vorgegebener Algorithmen erfolgen.

Entscheidungen zu fach- und/oder fächerübergreifenden Fragen: /

Jahrgangsstufe 9

Unterrichtsvorhaben III: Wie funktioniert ein Computer eigentlich? – Assembler und logische Schaltungen

Schwerpunkte der Kompetenzentwicklung – Übergeordnete Kompetenzerwartungen:

Die Schülerinnen und Schüler

- analysieren und beschreiben informatische Sachverhalte (A),
- bewerten mögliche Auswirkungen des Einsatzes von Informatiksystemen (A),
- strukturieren informatische Sachverhalte (MI),
- analysieren Modelle und Implementierungen (MI),
- entwickeln informatische Modelle zu gegebenen Problemstellungen (MI),
- beurteilen Modelle und Implementierungen hinsichtlich der Lösung einer Problemstellung (MI),
- veranschaulichen informatische Sachverhalte (DI),

- interpretieren Ergebnisse von Implementierungen (DI),
- stellen informatische Sachverhalte unter Verwendung von Fachbegriffen dar (KK).

Inhaltsfelder: Information und Daten; Informatiksysteme; Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- ♦ Erfassung, Verarbeitung und Verwaltung von Daten
- ♦ Anwendung von Informatiksystemen
- ♦ Assembler
- ♦ Logische Schaltungen
- ♦ Informatiksysteme in der Lebens- und Berufswelt

Konkretisierte Kompetenzerwartungen:

Die Schülerinnen und Schüler

- interpretieren Daten aus dem Ergebnis eines Verarbeitungsprozesses (DI),
- identifizieren für (vernetzte) Informatiksysteme kriteriengeleitet Anwendungsbereiche in der Lebens- und Berufswelt (A),
- erstellen und simulieren logische Schaltungen mithilfe digitaler Werkzeuge (MI),
- bewerten eine logische Schaltung hinsichtlich ihrer Funktionalität (A),
- diskutieren Auswirkungen des Einsatzes von Informatiksystemen an ausgewählten Beispielen aus der Berufswelt (A/KK). (BNE - 9)

Zeitbedarf: ca. 40 Ustd.

Ggf. Absprachen zur Leistungsüberprüfung: /

Verbindliche Hinweise und Absprachen zu diesem Unterrichtsvorhaben / Umsetzung:

Zunächst wird in das EVA-System eingeführt und die SuS erarbeiten sich Grundbestandteile (Hardware) eines PCs. In diesem Zusammenhang erarbeiten sich die SuS zudem die Von-Neumann-Rechnerarchitektur.

Mithilfe des „Bonsai“- Modellassemblers blicken die SuS auf die „Schnittstelle“ zwischen Soft- und Hardware und lernen an Beispielen die Bedeutung des Binärsystems für die (auch technische) Informatik kennen.

Mithilfe der Simulationssoftware LogicSim o.ä. für logische Schaltungen, untersuchen die Lernenden die Funktion der grundlegenden Gatter AND, OR, XOR und NOT sowie NAND und NOR. In einfachen Anwendungskontexten werden Schalttabellen bzw. Schaltungen entwickelt und ineinander überführt. Weiter werden logische Schaltungen hinsichtlich ihrer Funktionalität getestet und bewertet und Ausgaben von Schaltnetzen interpretiert. Schaltungen für verschiedene Steuerungen (z.B. Türöffner, Fahrstühle Beleuchtungen, Zähler, Sonnenschutzsysteme, Heizungsregler, Bahn- oder Flugsicherungssysteme) werden als Ausgangspunkte genutzt, um kriterien geleitet Anwendungsbereiche für einfache und vernetzte Informatiksysteme in der Lebens- und Berufswelt zu identifizieren und an ausgewählten Beispielen aus der Berufswelt die Auswirkungen des Einsatzes von Informatiksystemen auch in Hinblick auf eine nachhaltige Entwicklung diskutiert. Um zu verdeutlichen, wie ein Rechenwerk

funktioniert, simulieren die Lernenden Halb- und Volladdierer und kombinieren diese optional zu einem 4-Bit-Addier- und Subtrahierwerk.

Jahrgangsstufe 10

Unterrichtsvorhaben IV: Top Secret! – Kryptologie

Schwerpunkte der Kompetenzentwicklung – Übergeordnete Kompetenzerwartungen:

Die Schülerinnen und Schüler

- bewerten informatische Sachverhalte kriteriengeleitet, (A),
- entwickeln Handlungsstrategien für informatische Fragestellungen (A),
- strukturieren informatische Sachverhalte (MI),
- wenden ein informatisches Verfahren zur Lösung eines Problems an (MI).

Inhaltsfelder: Information und Daten; Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- ♦ Daten und ihre Codierung
- ♦ Verschlüsselungsverfahren
- ♦ Erfassung, Verarbeitung und Verwaltung von Daten
- ♦ Datenschutz und Datensicherheit

Konkretisierte Kompetenzerwartungen:

Die Schülerinnen und Schüler

- verwenden Substitutionsverfahren als Möglichkeit der Verschlüsselung (MI),
- beurteilen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (A), (MKR 1.4)
- erläutern die Prinzipien der Datensicherheit (Vertraulichkeit, Integrität und Verfügbarkeit) und berücksichtigen diese beim Umgang mit Daten (A), (MKR 1.4)
- entwickeln kriteriengeleitet Handlungsoptionen für den Umgang mit eigenen und fremden Daten (A).

Zeitbedarf: ca. 20 Ustd.

Ggf. Absprachen zur Leistungsüberprüfung: /

Verbindliche Hinweise und Absprachen zu diesem Unterrichtsvorhaben / Umsetzung:

Zunächst können die Themenbereiche Sicherheitsprobleme und Sicherheitsziele im Bereich der digitalen Kommunikation an aktuellen Beispielen beleuchtet werden (z. B. Phishing-Mails, Hackerangriffe). Die Sicherheitsziele „Vertraulichkeit, Integrität und Verfügbarkeit“ werden

hierbei zu zentralen Unterrichtsthemen und von den Schülerinnen und Schülern erläutert. Weiterhin wird das Bewusstsein dafür geschärft, wie privat oder öffentlich Nachrichten in sozialen Medien, in E-Mails oder auf anderen Internetplattformen sind. Fragestellungen können dabei z.B. sein: Kann jemand außer dem Empfänger meine E-Mails oder Chatnachrichten lesen? Wer kann das? Handlungsoptionen für den Umgang mit eigenen und fremden Daten werden entwickelt. Anschließend beschäftigen sich die Schülerinnen und Schüler mit kryptographischen Verfahren, um Botschaften zu verschlüsseln. Ein einfaches Beispiel dafür bietet der Cäsar-Algorithmus als Substitutionsverfahren. Die Beurteilung dieses Verschlüsselungsverfahrens unter Berücksichtigung einer möglichen Mustererkennung oder Ermittlung des Schlüssels durch eine Häufigkeitsanalyse führt zum Wunsch nach einem polyalphabetischen Chiffrierverfahren. Das Vigenère-Verfahren wird eingeführt und angewendet. Auch dieses Verfahren wird unter Berücksichtigung einer möglichen Mustererkennung oder Ermittlung des Schlüssels beurteilt. Weitere Aspekte, die für die Beurteilung eine Rolle spielen, sind das Verhältnis der Länge des verwendeten Schlüssels zum verschlüsselten Text, sowie die Notwendigkeit den Schlüssel zu übermitteln.

Entscheidungen zu fach- und/oder fächerübergreifenden Fragen: /

Jahrgangsstufe

10

Unterrichtsvorhaben V: Wir präsentieren uns im Internet – Aufbau und Struktur von Webseiten

Schwerpunkte der Kompetenzentwicklung – Übergeordnete Kompetenzerwartungen:

Die Schülerinnen und Schüler

- analysieren und beschreiben informatische Sachverhalte (A),
- strukturieren informatische Sachverhalte (MI),
- interpretieren unterschiedliche Darstellungen von informatischen Sachverhalten (DI),
- erläutern adressatengerecht informatische Sachverhalte (KK),
- stellen informatische Sachverhalte unter Verwendung von Fachbegriffen dar (KK),
- kooperieren im Rahmen des projektorientierten Arbeitens (KK), (MKR 3.1)
- planen die Dokumentation und Präsentation ihrer Vorgehensweise und Arbeitsergebnisse eigenständig (KK).

Inhaltsfelder: Information und Daten; Automaten und formale Sprachen; Informatiksysteme; Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- ♦ Daten und ihre Codierung
- ♦ Erstellung und Analyse von Quelltexten
- ♦ Anwendung von Informatiksystemen
- ♦ Datenschutz und Datensicherheit

Konkretisierte Kompetenzerwartungen:

Die Schülerinnen und Schüler

- interpretieren Daten aus dem Ergebnis eines Verarbeitungsprozesses (DI),
- beschreiben an ausgewählten Beispielen das Codierungsprinzip von Pixel- und Vektorgrafiken (KK),
- analysieren Quelltexte auf syntaktische Korrektheit (A/MI),
- erstellen syntaktisch korrekte Quelltexte in einer geeigneten Dokumentenbeschreibungssprache und in einer Programmiersprache (MI),
- wenden zielgerichtet Prinzipien der strukturierten Dateiverwaltung an (MI), (MKR 1.3)
- bewerten verschiedene Lizenzmodelle im Hinblick auf Weiterentwicklung und Nutzung digitaler Produkte (A), (MKR 4.4)
- entwickeln kriteriengeleitet Handlungsoptionen für den Umgang mit eigenen und fremden Daten (A). (MKR 1.3, 1.4)

Zeitbedarf: ca. 30 Ustd (+ ca. 6 Ustd. für die Präsentationen / Kolloquien)

Ggf. Absprachen zur Leistungsüberprüfung: Die SuS implementieren im Rahmen einer Projektarbeit eine eigene Website. Die Dokumentation des Projekts sowie die Präsentation samt Kolloquium ersetzt die zweite Arbeit im ersten Halbjahr.

Verbindliche Hinweise und Absprachen zu diesem Unterrichtsvorhaben / Umsetzung: Um den Schülerinnen und Schülern eine alltagsrelevante Anknüpfung zu ermöglichen und eine hohe Motivation zu erzeugen, ist dieses Unterrichtsvorhaben projektartig angelegt. Am Ende der Reihe steht eine Webseite als individuelles Produkt der Schülerinnen und Schüler. Zunächst müssen jedoch die Grundlagen der Beschreibung von Dokumenten und die Formatierung und Aufbereitung von Daten mittels Auszeichnungen eingeführt werden. Dazu können neben den Auszeichnungen selbst auch Formatierungsmöglichkeiten mit CSS genutzt werden. Dokumentenbeschreibungssprachen bieten aufgrund der breiten Anwendungsszenarien und des Sprachumfangs trotz eines einfachen Einstiegs eine Vielzahl individueller Differenzierungs- und Vertiefungsmöglichkeiten. Als grafische Elemente können auch Bilder eingebunden werden. Ein Vergleich von Pixel und Vektorgrafiken bietet sich an. Die Schülerinnen und Schüler präsentieren sich in diesem Alter vielleicht schon im Netz, in jedem Fall haben sie in Ihrem Alltag vielfältige Berührungspunkte mit Webseiten und Apps. So ergibt sich einerseits eine direkte Anknüpfung an den Alltag als auch die Notwendigkeit auf Möglichkeiten und Pflichten bei der digitalen Veröffentlichung von Daten einzugehen. Dabei stehen nicht nur soziale Regeln (Netiquette, Regeln zur Veröffentlichung, Anonymität im Netz, Barrierefreiheit), sondern auch rechtliche Pflichten (Datenschutz, Urheberrecht, Lizenzen) im Fokus.

Entscheidungen zu fach- und/oder fächerübergreifenden Fragen: /

Schwerpunkte der Kompetenzentwicklung – Übergeordnete Kompetenzerwartungen:

Die Schülerinnen und Schüler

- analysieren und beschreiben informatische Sachverhalte (A),
- bewerten mögliche Auswirkungen des Einsatzes von Informatiksystemen (A),
- entwickeln Handlungsstrategien für informatische Fragestellungen (A),
- erläutern adressatengerecht informatische Sachverhalte (KK),
- stellen informatische Sachverhalte unter Verwendung von Fachbegriffen dar (KK).

Inhaltsfelder: Information und Daten; Informatiksysteme; Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- ♦ überwachtes Lernen
- ♦ unüberwachtes Lernen
- ♦ bestärkendes Lernen

Konkretisierte Kompetenzerwartungen:

Die Schülerinnen und Schüler

- beschreiben Anwendungsbeispiele künstlicher Intelligenz zum überwachtem, unüberwachten und bestärkenden Lernen (KK),
- beschreiben die grundlegende Funktionsweise maschinellen Lernens (überwacht, unüberwacht, bestärkend) in verschiedenen Anwendungsbeispielen (KK), (MKR 6.1)
- ordnen begründet die Methoden des maschinellen Lernens (überwachtes Lernen, unüberwachtes, bestärkendes Lernen) verschiedenen Anwendungsbeispielen zu (A),
- analysieren den Einfluss von Trainingsdaten auf die Ergebnisse eines Verfahrens maschinellen Lernens (A). (MKR 6.4)

Zeitbedarf: ca. 20 Ustd.

Ggf. Absprachen zur Leistungsüberprüfung: /

Verbindliche Hinweise und Absprachen zu diesem Unterrichtsvorhaben / Umsetzung:

Das Unterrichtsvorhaben knüpft an das Unterrichtsvorhaben zu Künstlicher Intelligenz in Jahrgang 6 an. Ausgehend von der Lebens- und Erfahrungswelt der Lerngruppen werden in der Klasse 10 Anwendungsbeispiele von KI-Systemen gesammelt, strukturiert und durch die Lehrkraft ergänzt. Für das überwachte Lernen werden die Grundideen aus der Klasse 6 zur Entwicklung eines Entscheidungsbaumes wiederholt und gefestigt (z. B. „Lebensmittel-KI). Dabei wird auch der Einfluss der Trainingsdaten auf die Ergebnisse analysiert. Die Grundidee des unüberwachten Lernens zum Clustern von Daten wird mithilfe der Unplugged-Aktivität „Goldrush“ eingeführt (vgl. <https://computingeducation.de/proj-snaip-B/>). Diese Grundidee kann später ggf. auf einen eigenen Datensatz angewandt werden. Die Grundidee des bestärkenden Lernens kann mithilfe der Unplugged-Aktivität „Mensch, Maschine!“-Spiel (vgl.

<https://www.prodabi.de/mensch-maschine-spiel/>) oder der interaktiven Webseite (vgl. <https://www.stefanseegerer.de/schlag-das-krokodil/>) eingeführt werden.

Im Verlauf der Unterrichtsreihe soll zudem auch auf aktuelle (gesellschaftliche) Entwicklungen, Chancen und Gefahren von Künstlicher Intelligenz eingegangen werden.

Entscheidungen zu fach- und/oder fächerübergreifenden Fragen: Je nach Kontext kann ggf. auf aktuelle politische Entwicklungen eingegangen werden, die im Wirtschaft-Politikunterricht thematisiert werden. Zudem werden die Erfahrungen der SuS aus dem Betriebspraktikum in der 10. Klasse dahingehend reflektiert, wie in den unterschiedlichen Berufsfeldern KI zum Einsatz kommt und das entsprechende Berufsfeld u.U. transformieren wird.

2.3 Unterrichtsvorhaben Sekundarstufe II

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

Im „Übersichtsraster Unterrichtsvorhaben“ 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. Exkursionen, 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 2.3.2) 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 2.3 bis 2.4 zu entnehmen sind.

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

2.3.1 Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

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

Einführungsphase

Unterrichtsvorhaben E-III

Thema:

Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 18 Stunden

Unterrichtsvorhaben E-IV

Thema:

Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anwendungsbeispiele

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 18 Stunden

Einführungsphase	
<u>Unterrichtsvorhaben E-V</u> Thema: <i>Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Algorithmen Inhaltliche Schwerpunkte: • Algorithmen zum Suchen und Sortieren • Analyse, Entwurf und Implementierung einfacher Algorithmen Zeitbedarf: 9 Stunden	<u>Unterrichtsvorhaben E-VI</u> Thema: <i>Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</i> Zentrale Kompetenzen: • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatik, Mensch und Gesellschaft • Informatiksysteme Inhaltliche Schwerpunkte: • Wirkungen der Automatisierung • Geschichte der automatischen Datenverarbeitung • Digitalisierung Zeitbedarf: 15 Stunden
Summe Einführungsphase: 74	

II) Qualifikationsphase (Q1 und Q2) – GRUNDKURS

Qualifikationsphase 1	
<u>Unterrichtsvorhaben Q1-I</u> Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen Zeitbedarf: 8 Stunden	<u>Unterrichtsvorhaben Q1-II</u> Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache Zeitbedarf: 20 Stunden

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 16 Stunden

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Sicherheit

Zeitbedarf: 20 Stunden

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

Qualifikationsphase 2

Unterrichtsvorhaben Q2-I

Thema:

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 24 Stunden

Unterrichtsvorhaben Q2-II

Thema:

Endliche Automaten und formale Sprachen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Endliche Automaten und formale Sprachen

Inhaltliche Schwerpunkte:

- Endliche Automaten
- Grammatiken regulärer Sprachen
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

Zeitbedarf: 20 Stunden

Qualifikationsphase 2	
<u>Unterrichtsvorhaben Q2-III</u> Thema: <i>Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</i> Zentrale Kompetenzen: • Argumentieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatiksysteme • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung Zeitbedarf: 12 Stunden	
Summe Qualifikationsphase 2: 56 Stunden	

II) Qualifikationsphase (Q1 und Q2) – LEISTUNGSKURS

Qualifikationsphase 1	
<u>Unterrichtsvorhaben Q1-I</u> Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung unter Berücksichtigung der Gestaltung einer Benutzungsoberfläche</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen Zeitbedarf: 15 Stunden	<u>Unterrichtsvorhaben Q1-II</u> Thema: <i>Modellierung und Implementierung von dynamischen linearen Datenstrukturen und von Anwendungen mit dynamischen linearen Datenstrukturen in kontextbezogenen Problemstellungen</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache Zeitbedarf: 25 Stunden

Qualifikationsphase 1

Unterrichtsvorhaben Q1-III

Thema:

Modellierung, Implementierung, Analyse und Beurteilung von Such- und Sortierv Verfahren unterschiedlicher Komplexitätsklassen in kontextbezogenen Problemstellungen

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 20 Stunden

Unterrichtsvorhaben Q1-IV

Thema:

Modellierung, Implementierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 20 Stunden

Qualifikationsphase 1

<u>Qualifikationsphase 1</u>	
<u>Unterrichtsvorhaben Q1-V</u> Thema: <i>Modellierung und Implementierung von dynamischen nicht-linearen Datenstrukturen und von Anwendungen mit dynamischen nicht-linearen Datenstrukturen in kontextbezogenen Problemstellungen.</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache Zeitbedarf: 40 Stunden	<u>Unterrichtsvorhaben Q1-VI</u> Thema: <i>Projektorientierte Softwareentwicklung am Beispiel einer Anwendung mit Datenbankanbindung</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit Zeitbedarf: 15 Stunden
Summe Qualifikationsphase 1: 135 Stunden	

Qualifikationsphase 2

<u>Unterrichtsvorhaben Q2-I</u> Thema: <i>Sicherheit und Datenschutz in Informatiksystemen sowie Grenzen und Auswirkungen der Automatisierung</i> Zentrale Kompetenzen: • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatiksysteme • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Nutzung von Informatiksystemen • Sicherheit • Wirkungen der Automatisierung • Grenzen der Automatisierung Zeitbedarf: 20 Stunden	<u>Unterrichtsvorhaben Q2-II</u> Thema: <i>Grundlagen der Netzwerkkommunikation sowie Modellierung und Implementierung von Client-Server-Anwendungen in kontextbezogenen Problemstellungen</i> Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Algorithmen • Informatiksysteme Inhaltliche Schwerpunkte: • Algorithmen in ausgewählten Kontexten • Einzelrechner und Rechnernetzwerke • Nutzung von Informatiksystemen Zeitbedarf: 20 Stunden
---	--

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Grundlagen von Automaten und formalen Sprachen sowie die Modellierung und Implementierung eines Parsers zu einer formalen Sprache

Zentrale Kompetenzen:

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

Inhaltsfelder:

- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf: 30 Stunden

Unterrichtsvorhaben Q2-IV

Thema:

Prinzipielle Arbeitsweise eines Computers sowie Modellierung und Implementierung eines Scanners, Parsers und Interpreters für eine einfache maschinennahe Programmiersprache

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Scanner, Parser und Interpreter für eine reguläre Sprache
- Einzelrechner und Rechnernetzwerke

Zeitbedarf: 12 Stunden

Qualifikationsphase 2

Unterrichtsvorhaben Q2-V

Thema:

*Entwicklung eines Netzwerkspiels mit
Durchführung eines vollständigen
Softwareentwicklungszyklus*

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und
Implementierung von Algorithmen
- Syntax und Semantik einer
Programmiersprache
- Einzelrechner und Rechnernetzwerke
- Nutzung von Informatiksystemen

Zeitbedarf: 15 Stunden

Summe Qualifikationsphase 2: 97 Stunden

2.3.2 Konkretisierte Unterrichtsvorhaben

Im Folgenden werden die im *Unterkapitel 2.3.1* aufgeführten Unterrichtsvorhaben konkretisiert.

I) Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

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

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Information, deren Kodierung und Speicherung (a) Informatik als Wissenschaft der Verarbeitung von Informationen (b) Daten und ihre Strukturierung (c) Möglichkeiten zur Darstellung von Informationen	Die Schülerinnen und Schüler • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	<i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekannten Zeichensätzen (z.B. Wingdings) <i>Beispiel:</i> Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken
2. Informations- und Datenübermittlung in Netzen (a) „Sender-Empfänger-Modell“ (b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)		<i>Beispiel:</i> Rollenspiel zur Paketvermittlung im Internet Schülerinnen und Schüler übernehmen die Rollen von Clients und Routern. Sie schicken spielerisch Informationen auf Karten von einem Schüler-Client zum anderen. Jede Schülerin und jeder Schüler hat eine Adresse, jeder Router

(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll) (d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet		darüber hinaus eine Routingtabelle. Mit Hilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weiter vermittelt wird.
3. Aufbau informatischer Systeme (a) Prinzipien und Komponenten der „Von-Neumann-Architektur“ (b) Erarbeitung des EVA-Prinzips auf Grundlage der „Von-Neumann-Architektur“		<i>Beispiel:</i> Demonstrationshardware Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an. <i>Beispiel:</i> Rollenspiel zur „Von-Neumann-Architektur“

Unterrichtsvorhaben EF-II

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

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

Vorhabenbezogene Konkretisierung:

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

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

Im Anschluss wird mit der Realisierung erster Projekte begonnen. Die von der verwendeten Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

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

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Identifikation von Objekten (a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt. (b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen. (c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst. (d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters	Die Schülerinnen und Schüler - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - stellen die Kommunikation zwischen Objekten grafisch dar (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - stellen den Zustand eines Objekts dar (D).	<i>Beispiel:</i> Vogelschwarm Schülerinnen und Schüler betrachten einen Vogelschwarm als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Allgemeine Objektorientierung
2. Analyse von Klassen didaktischer Lernumgebungen (a) Objektorientierte Programmierung (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen: Analyse,		<i>Materialien:</i> Dokumentation der didaktischen Bibliothek GLOOP

Modellierung und Implementation)		
(b) Teilanalyse vorgegebener Klassen		
3. Implementierung grundlegender Aktionen in der verwendeten didaktischen Bibliothek (a) Grundaufbau einer Java-Klasse (b) Deklaration und Initialisierung von Objekten (c) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften		<i>Beispiel:</i> Olympische Ringe Die Schülerinnen und Schüler bilden das Emblem der olympischen Spiele mit Hilfe der verwendeten Bibliothek nach. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Sequenzielle Programmierung

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer 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	Beispiele, Medien, Materialien
1. Bewegungsanimationen am Beispiel einfacher grafischer Objekte (a) Kontinuierliche Verschiebung eines Objektes mit Hilfe einer Schleife (While-Schleife) (b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife (c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen (d) Berechnung von Abständen zwischen Objekten mit Hilfsvariablen (e) Meldungen zur Kollision zweier Objekte mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)	Die Schülerinnen und Schüler • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • 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), • modifizieren einfache Algorithmen und Programme (I),	<i>Beispiel: Zeichenprogramm</i> Entwicklung eines Zeichenprogramms mit vielfältigen Funktionen <i>Beispiel: Schatzsuche</i> Ein zufällig auf dem Bildschirm versteckter Schatz muss vom Benutzer gefunden werden. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Kontrollstrukturen
2. Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte		<i>Beispiel: Hubschrauberlandeplatz</i>

(a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife) (b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays) (c) Verwendung von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden	• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	Die Schülerinnen und Schüler realisieren einen runden Hubschrauberlandeplatz und eine Reihe von Landemarkierungen, die in einem Feld verwaltet werden. Mit Hilfe der Landemarkierungen werden verschiedene Lauflichter realisiert. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Kontrollstrukturen
3. Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte (a) Modellierung eines Simulationsprogramms mit eigenen Klassen (b) Implementierung und Anwendung eigener Methoden mit und ohne Parameterübergabe		<i>Beispiel: Kerzensimulation</i> Die Schülerinnen und Schüler modellieren und erstellen eine Klasse, mit deren Hilfe Kerzen simuliert werden können. Eine Kerze kann angezündet und gelöscht werden. Abgesehen davon brennen Kerzen abhängig von ihrer Dicke unterschiedlich schnell ab.

(c) Realisierung von Zustandsvariablen (d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten (e) Vertiefung: Weitere Projekte		<i>Beispiel: CAD-Programm</i> Die Schülerinnen und Schüler modellieren und erstellen eigene Klassen, mit deren Hilfe sich Grundrisse von Gebäuden realisieren lassen. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Eigene Klassen
---	--	--

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand *lebensnaher Anwendungsbeispiele*

Leitfrage: *Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?*

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	Die Schülerinnen und Schüler • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • 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),	<i>Beispiel: Teletennis</i> Die Schülerinnen und Schüler durchlaufen den Prozess der Programmentwicklung anhand eines Spiels, bei dem ein Ball mit Hilfe zweier Schläger in einem begrenzten Spielfeld gehalten werden muss. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Assoziationen Informationsblatt: Implementationsdiagramme
2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden (b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendiagramm		

<i>(d) Erstellung von Implementationsdiagrammen als Vorbereitung für die Programmierung</i>	• modellieren Klassen unter Verwendung von Vererbung (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • modifizieren einfache Algorithmen und Programme (I), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Vererbung
3. Programmierung anhand von Implementationsdiagrammen (a) Klassen werden in Java-Quellcode umgesetzt (b) Das Geheimnisprinzip wird umgesetzt <i>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</i>		
4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet		<i>Beispiel: Erweiterung des Programms Teletennis</i> <i>Materialien:</i>

(b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert		Ergänzungsmaterialien zum Lehrplannavigator – Vererbung
--	--	---

Unterrichtsvorhaben EF-V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

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

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

Des Weiteren soll das Prinzip der *binären Suche* behandelt und nach Effizienzgesichtspunkten untersucht werden.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Explorative Erarbeitung eines Sortierverfahrens (a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.) (b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus (c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler	Die Schülerinnen und Schüler • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), • entwerfen einen weiteren Algorithmus zum Sortieren (M), • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).	<i>Beispiel:</i> Sortieren mit Waage Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können. <i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: abgerufen: 30. 03. 2014
		<i>Beispiele:</i> Sortieren durch Auswählen,

2. Systematisierung von Algorithmen und Effizienzbetrachtungen (a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen) (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs (e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)		Sortieren durch Vertauschen, Quicksort Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird. <i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: abgerufen: 30. 03. 2014
3. Binäre Suche auf sortierten Daten (a) Suchaufgaben im Alltag und im Kontext informatischer Systeme		<i>Beispiel:</i> Simulationsspiel zur binären Suche nach Tischtennisbällen

(b)Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche (c) Effizienzbetrachtungen zur binären Suche		Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden. <i>Materialien:</i> Computer science unplugged – Searching Algorithms, URL: , abgerufen: 30. 03. 2014
---	--	--

Unterrichtsvorhaben EF-VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

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

Vorhabenbezogene Konkretisierung:

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

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

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler (a) Mögliche Themen zur Erarbeitung in Kleingruppen: • „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“ • „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ (b) Vorstellung und Diskussion durch Schülerinnen und Schüler	Die Schülerinnen und Schüler • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).	<i>Beispiel:</i> Ausstellung zu informatischen Themen Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können. <i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.

2. Vertiefung des Themas Datenschutz (a) Erarbeitung grundlegender Begriffe des Datenschutzes (b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler (c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“		<i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung. <i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz (Download EF-VI.1)
---	--	---

II) Qualifikationsphase - GRUNDKURS

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben Q1-I

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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

Vorhabenbezogenen Konkretisierung:

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

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

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels (a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (f) Implementierung der Anwendung oder von Teilen der Anwendung	Die Schülerinnen und Schüler • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der	<i>Beispiel: Wetthuepfen</i> Für ein Wetthüpfen zwischen einem Hasen, einem Hund und einem Vogel werden die Tiere gezeichnet. Alle Tiere springen wiederholt nach links. Die Höhe und Weite jedes Hüpfers ist zufällig. Evtl. marschieren sie anschließend hintereinander her. oder <i>Beispiel: Tannenbaum</i> Ein Tannenbaum soll mit verschiedenen Arten von Schmuckstücken versehen werden, die durch unterschiedliche geometrische Objekte dargestellt werden. Es gibt Kugeln, Päckchen in der Form von Würfeln und Zuckerringe in Form von Toren. Ein Prototyp, der bereits mit Kugeln geschmückt werden kann, kann zur Verfügung gestellt werden. Da alle Schmuckstücke über die Funktion des Auf- und Abschmückens verfügen sollen, liegt es nahe, dass entsprechende Methoden in einer gemeinsamen Oberklasse realisiert werden. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.1-Wiederholung (Download Q1-I.1)

	Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D)	
--	---	--

Unterrichtsvorhaben Q1-II

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

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

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

	• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).	
2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung der Funktionalität der Klasse Stack (c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack		<i>Beispiel: Heftstapel</i> In einem Heftstapel soll das Heft einer Schülerin gefunden werden. oder <i>Beispiel: Kisten stapeln</i> In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.
3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List (a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen (b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.		<i>Beispiel: Abfahrtslauf</i> Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen

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

anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.

Beispiel: Autos an einer Ampel zur Zufahrtsregelung

Es soll eine Ampel zur Zufahrtsregelung in Java simuliert werden. An einem geradlinigen, senkrecht von unten nach oben verlaufenden Straßenstück, das von Autos nur einspurig in eine Richtung befahren werden kann, ist ein Haltepunkt markiert, an dem die Ampel steht. Bei einem Klick auf eine Schaltfläche mit der Aufschrift „Heranfahren“ soll ein neues Auto an den Haltepunkt heranfahren bzw. bis an das letzte Auto, das vor dem Haltepunkt wartet. Grünphasen der Ampel werden durch einen Klick auf eine Schaltfläche mit der Aufschrift „Weiterfahren“ simuliert. In jeder Grünphase darf jeweils nur ein Auto weiterfahren. Die anderen Autos rücken nach.

Materialien:

		Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen
--	--	---

Unterrichtsvorhaben Q1-III

Thema: Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)	Die Schülerinnen und Schüler • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der	<i>Beispiel:</i> Karteiverwaltung Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden. oder <i>Beispiel:</i> Bundesjugendspiele Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken. Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin heraussuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren

2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)	Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	<i>Beispiel:</i> Karteiverwaltung (s.o.) oder <i>Beispiel:</i> Bundesjugendspiele (s.o.) <i>Materialien:</i> (s.o.)
3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen (a) Grafische Veranschaulichung der Sortierverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarfes bei beiden Sortierverfahren (c) Beurteilung der Effizienz der beiden Sortierverfahren		<i>Beispiel:</i> Karteiverwaltung (s.o.) oder <i>Beispiel:</i> Bundesjugendspiele (s.o.) <i>Materialien:</i> (s.o.)

Unterrichtsvorhaben Q1-IV

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Nutzung von relationalen Datenbanken (a) Aufbau von Datenbanken und Grundbegriffe - Entwicklung von Fragestellungen zur vorhandenen Datenbank - Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema (b) SQL-Abfragen - Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle - Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <, >, <=, >=, <=, LIKE, BETWEEN, IN, IS NULL)	Die Schülerinnen und Schüler - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), - analysieren und erläutern eine Datenbankmodellierung (A), - erläutern die Eigenschaften normalisierter Datenbankschemata (A), - bestimmen Primär- und Sekundärschlüssel (M), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - modifizieren eine Datenbankmodellierung (M), - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - bestimmen Primär- und Sekundärschlüssel (M),	<i>Beispiel:</i> Fitnessstudio [derzeit in Überarbeitung] <i>Beispiel:</i> Schulbuchausleihe Unter (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.

(c) Vertiefung an einem weiteren Datenbankbeispiel	• überführen Datenbankschemata in vorgegebene Normalformen (M), • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einen	
2. Modellierung von relationalen Datenbanken (a) Entity-Relationship-Diagramm • Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms • Erläuterung und Modifizierung einer Datenbankmodellierung (b) Entwicklung einer Datenbank aus einem Datenbankentwurf • Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln (c) Redundanz, Konsistenz und Normalformen • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der	• Datenbanksystem zu extrahieren (I), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	<i>Beispiel: Fahrradverleih</i> Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von <i>BTR</i> können CityBikes, Treckingräder und Mountainbikes ausleihen. <i>Beispiel: Reederei</i> Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird. <i>Beispiel: Buchungssystem</i> In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist.

1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)		Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden. Unter (abgerufen: 30.03. 2014) findet man ein freies Online-Buchungssystem inklusive Demo, anhand derer man erläutern kann, worum es in dem Projekt geht. <i>Beispiel: Schulverwaltung</i> In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.
--	--	--

Unterrichtsvorhaben Q1-V

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

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

2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).	<i>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz</i>
---	--	---

Unterrichtsvorhaben Q2-I:

Thema: Endliche Automaten und formale Sprachen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Endliche Automaten (a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten (b) Untersuchung, Darstellung und Entwicklung endlicher Automaten	Die Schülerinnen und Schüler 1. analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), 2. analysieren und erläutern Grammatiken regulärer Sprachen (A), 3. zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), 4. ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), 5. entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), 6. entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),	<i>Beispiele:</i> Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen Modellierungssoftware JFlap
2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen		<i>Beispiele:</i>

(a) Erarbeitung der formalen Darstellung regulärer Grammatiken (b) Untersuchung, Modifikation und Entwicklung von Grammatiken (c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden (d) Entwicklung regulärer Grammatiken zu endlichen Automaten	7. entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), 8. entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), 9. modifizieren Grammatiken regulärer Sprachen (M), 10. entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), 11. stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), 12. ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). 13. beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).	reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik <i>Materialien:</i> (s.o.)
3. Grenzen endlicher Automaten		<i>Beispiele:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$

Unterrichtsvorhaben Q2-II:

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

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

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

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

Zeitbedarf: 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Analyse von Baumstrukturen in verschiedenen Kontexten (a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) (b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten	Die Schülerinnen und Schüler • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),	<i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht. oder <i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat. <i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i> <i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten)

	• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • 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),	Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) <i>oder</i> <i>Beispiel: Entscheidungsbäume</i> Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum (Beispiel: Akinator). <i>oder</i>
--	---	---

	• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	<i>Beispiel:</i> Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum
2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree <i>(a)</i> Analyse der Problemstellung, Ermittlung von Objekten, ihren		<i>Beispiel:</i> Informatikerbaum als binärer Baum In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem

Eigenschaften und Operationen im Anwendungskontext (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms (c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen (d) Implementierung der Anwendung oder von Teilen der Anwendung (e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf		Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt: • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum
3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree		<i>Beispiel:</i> Informatikerbaum als <i>Suchbaum</i> In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in

(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften (c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation (d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums		dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt: • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		<i>Beispiel:</i> Codierungsbäume (s.o.) oder Huffman-Codierung <i>oder</i> <i>Beispiel:</i> Buchindex

		Es soll eine Anwendung entwickelt werden, die anhand von Stichworten und zugehörigen Seitenzahlen ein Stichwortregister erstellt. Da die Stichwörter bei der Analyse des Buches häufig gesucht werden müssen, werden sie in der Klasse Buchindex als Suchbaum (Objekt der Klasse BinarySearchTree) verwaltet. Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) <i>oder</i> <i>Beispiel:</i> Entscheidungsbäume (s.o.) <i>oder</i> <i>Beispiel:</i> Termbaum (s.o.) <i>oder</i> <i>Beispiel:</i> Ahnenbaum (s.o.) <i>Materialien:</i>
--	--	---

		Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Anwendung Binärbaum
--	--	--

Unterrichtsvorhaben Q2-III:

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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

Vorhabenbezogene Konkretisierung:

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

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

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU,	Die Schülerinnen und Schüler	<i>Beispiel:</i> Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell

Rechenwerk, Steuerwerk, Register und Hauptspeicher b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	1. erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), 2. untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung
2. Grenzen der Automatisierbarkeit a) Vorstellung des Halteproblems b) Unlösbarkeit des Halteproblems c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen		<i>Beispiel:</i> Halteproblem <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem

Unterrichtsvorhaben Q2-IV:

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

III) Qualifikationsphase - LEISTUNGSKURS

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben Q1-I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung unter Berücksichtigung der Gestaltung einer Benutzungsoberfläche

Leitfragen: *Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen? Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen? Wie kann ein Softwareprojekt mit angemessener Benutzungsoberfläche arbeitsteilig entwickelt und getestet werden?*

Vorhabenbezogenen Konkretisierung:

Zu einer Problemstellung in einem Anwendungskontext soll eine Java-Anwendung entwickelt werden. Diese Anwendung soll aus einer Benutzungsoberfläche und einer davon getrennten Verarbeitung der mit der Oberfläche eingegebenen Daten bestehen. Auf diese Weise werden nicht nur Grundideen der Objektorientierung wiederholt, sondern auch erste Schritte zur Thematisierung des MVC-Prinzips realisiert.

Die Schülerinnen und Schüler analysieren zu Beginn die gegebene Problemstellung und formulieren Anforderungen an die zu entwickelnde Java-Anwendung. Anschließend wird eine Benutzungsoberfläche nach softwareergonomischen Kriterien konzipiert und mit einem GUI-Builder umgesetzt.

Mit Blick auf die fertige Benutzungsoberfläche wird eine Objektorientierte Modellierung entwickelt und in Form eines Implementationsdiagramms fixiert. Wesentliche Klassen werden von Schülerinnen und Schülern dokumentiert. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der Objektorientierten Programmierung wiederholt. Eine Implementierung der Java-Anwendung kann arbeitsteilig erfolgen. Zum Schluss wird das fertige Projekt mit Hilfe einer Testanwendung getestet.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Entwicklung eines Anforderungskatalogs (a) Analyse der Problemstellung (b) Entwicklung von Anwendungsfällen (use cases) (c) Formulierung eines Anforderungskatalogs	Die Schülerinnen und Schüler • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der	<i>Beispiel:</i> Taschenrechner Schülerinnen und Schüler entwickeln eine Taschenrechneranwendung. Dabei kann es sich um einen Rechner mit einfachen Grundrechenarten bis hin zu einem wissenschaftlichen Taschenrechner mit umfangreicheren Funktionen handeln. Die Anforderungen sind von den Schülerinnen und Schülern selbst zu definieren.
2. Entwurf einer grafischen Benutzungsoberfläche (a) Erarbeitung softwareergonomischer Prinzipien für Benutzungsoberflächen (b) Entwurf von Prototypen für eine Benutzungsoberfläche unter Berücksichtigung des Anforderungskatalogs (c) Einarbeitung in die Funktionsweise eines GUI-Builders (d) Realisierung einer Benutzungsoberfläche mit einem GUI-Builder	(continued from previous row)	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.I

3. Klassenmodellierung und Visualisierung von Objektkommunikation (a) Entwurf eines Implementationsdiagramms unter Berücksichtigung des Anforderungskatalogs und der Benutzungsoberfläche (b) Grafische Darstellung der Kommunikation zwischen mindestens zwei Objekten (c) Dokumentation von Klassen	Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen und mithilfe von Testanwendungen(I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K).	
4. Projektentwicklung und Test (ggf. arbeitsteilig) (a) Implementierung der Anwendung mit grafischer Benutzungsoberfläche (b) Test einzelner Klassen der Gesamtanwendung mit Hilfe von Testanwendungen		<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.I

Unterrichtsvorhaben Q1-II:

Thema: Modellierung und Implementierung von dynamischen linearen Datenstrukturen und von Anwendungen mit dynamischen linearen Datenstrukturen in kontextbezogenen Problemstellungen

Leitfragen: *Wie können beliebig viele linear angeordnete Daten in unterschiedlichen Anwendungskontexten problemgerecht verwaltet werden?*

Vorhabenbezogenen Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem First-In-First-Out-Prinzip verwaltet werden, wird der Aufbau von einer geeigneten Datenstruktur entwickelt und implementiert. Es werden unterschiedliche Implementationsmöglichkeiten verglichen. Anschließend werden die für die Anwendung notwendigen Klassen modelliert und implementiert. Die Klasse Queue wird als Alternative zur selbstentwickelten Datenstruktur vorgestellt. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben.

Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Eine der beiden Klassen zu den linearen Strukturen sollte vollständig modelliert, implementiert und dokumentiert werden.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird ein Anwendungskontext vorgestellt, bei dem die Anwendung der Klassen Queue und Stack nicht problemgerecht ist. Die Klasse List, bei der das Einfügen an beliebiger Stelle einer sequentiell angeordneten Struktur möglich ist, wird eingeführt und in einem Anwendungskontext verwendet.

In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Zeitbedarf: 25 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Modellierung und Implementation einer Datenstruktur Schlange , die die Daten nach dem FIFO-Prinzip im Anwendungskontext verwaltet.	Die Schülerinnen und Schüler - modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), - analysieren und erläutern Algorithmen und Programme (A), - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M),	<i>Beispiel:</i> (Schlange) Patientenwarteschlange Sobald eine Patientin oder ein Patient in einer Arztpraxis eintrifft, werden Name und Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ von Patientinnen und Patienten sowie das „Entfernen“ von Patientinnen und Patienten, nachdem sie zur Behandlung gerufen wurden. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Erster Lösungsansatz: Die FIFO-Datenstruktur wird problembezogen modelliert und implementiert. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II

	• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),	
2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung der Funktionalität der Klasse Queue und des Konzepts der parametrisierten Klassen Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue	• 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 und mit Hilfe von Testanwendungen (I),	Zweiter Lösungsansatz: Bei der Modellierung und Implementation wird der generische Datentyp Queue verwendet. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II

3. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack (a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen (b) Erarbeitung der Funktionalität der Klasse Stack (c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack	stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).	<i>Beispiel:</i> (Stapel) Handdechiffrierer mit Undo-Operation. In einem Text sind Buchstaben vertauscht worden, er wird so zu einem Geheimtext. Dieser Text wird in ein Softwareprodukt kopiert und schrittweise dechiffriert. Es können für jedes Zeichen Ersetzungsvorschläge gemacht werden und die Software setzt diese Ersetzung im gesamten Text um. Am Kontrolltext bemerkt man, ob die Ersetzungen sinnvoll waren. Falls man einen Irrweg gegangen ist, kann man die Ersetzungen bis zu dem gewünschten Punkt schrittweise rückgängig machen. Die Modellierung und die Implementation erfolgen mit dem generischen Datentyp Stack. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II
---	--	---

4. Die Datenstruktur Lineare Liste im Anwendungskontext unter Nutzung der Klasse List (a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen (b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List		<i>Beispiel:</i> (Liste) Todo-Liste Es werden Aufgaben verwaltet, die in eine vom Benutzer gewünschte Reihenfolge angeordnet werden. Der Benutzer kann in der Todo-Liste navigieren, bestehende Einträge ändern oder löschen, sowie vor oder hinter einem Eintrag neue Aufgaben einfügen. Ein Test, in dem alle Aufgaben aufgelistet sind, kann ebenfalls abgerufen werden. Die Modellierung und die Implementation erfolgt mit dem generischen Datentyp List. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.II
5. Vertiefung – Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext (a) Modellierung und Implementierung einer weiteren Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue (b) Modellierung und Implementierung einer weiteren Anwendung unter Verwendung		<i>Beispiele für Vertiefungen</i> <i>Beispiel:</i> (Schlange) Auslastungssimulation einer Arztpraxis. Bei einer Arztpraxis wird die Auslastung des Arztes während der Sprechstunde simuliert. Eingegeben werden die Öffnungszeiten und die voraussichtliche Anzahl der zu erwartenden Patienten. Die Behandlungsdauern der Patientinnen und Patienten werden zufällig gesetzt. Ergebnis der Simulation sind die Zeiten, die die Arztpraxis nicht ausgelastet ist und die Zeit, die am Ende der Sprechstunde notwendig

eines oder mehrerer Objekte der Klasse Stack (c) Modellierung und Implementierung einer weiteren kontextbezogenen Anwendung unter Verwendung der Klasse List		ist, um alle noch wartenden Patienten zu behandeln. <i>Beispiel:</i> (Stapel) Terme in Postfix-Notation Die sog. UPN (<i>umgekehrte polnische Notation</i>) bzw. <i>Postfix-Notation</i> eines Terms setzt den Operator hinter die Operanden. Um den Wert des Terms zu berechnen, kann ein Stack verwendet werden. <i>Beispiel:</i> (Liste) Verwaltung einer Todo-Liste mit Prioritäten (Todo-Listenmanager) Es werden Aufgaben verwaltet, die mit einer gewissen Priorität erledigt werden sollen. Es werden drei Prioritätsstufen unterschieden und die Aufgaben so angeordnet, dass sie in passender Reihenfolge stehen.
--	--	--

Unterrichtsvorhaben Q1-III:

Thema: Modellierung, Implementierung, Analyse und Beurteilung von Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen in kontextbezogenen Problemstellungen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Suchen von Daten in Listen und Arrays (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Speicherbedarf, Anzahl der Vergleiche)	Die Schülerinnen und Schüler - analysieren und erläutern Algorithmen und Programme (A), - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ , „Teilen und Herrschen“ und „Backtracking“(M),	<i>Beispiel:</i> Benutzerverwaltung für einen Fotokopierer Benutzerinnen und Benutzer des Kopierers geben eine vierstellige Ziffernkombination an, um sich zu legitimieren. Über diesen PIN-Code soll auch die Benutzerinnen und Benutzer des Kopierers identifizierbar sein, um auf sein Konto den Kopierverbrauch zu buchen. Die Benutzungsdaten werden in linearen Strukturen verwaltet, die nach den PIN-Nummern sortiert vorliegen. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.III - Suchen und Sortieren

2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierv Verfahren (a) Entwicklung und Implementierung eines einfachen Sortiervfahrens für eine Liste (b) Implementierung eines einfachen Sortiervfahrens für ein Feld (c) Entwicklung eines rekursiven Sortiervverfahren für ein Feld (z.B. Sortieren durch Mischen)	• modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortiervverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten)(I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I),	<i>Beispiel:</i> Benutzerverwaltung für einen Fotokopierer (s.o.) Die Benutzungsdaten sollen nach den Kriterien Kopierverbrauch, Namen und PIN-Nummern geordnet ausgegeben werden können. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.III - Suchen und Sortieren
3. Untersuchung der Effizienz der Sortiervverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen (a) Grafische Veranschaulichung der Sortiervverfahren (b) Untersuchung der Anzahl der Vergleichsoperationen bei beiden Sortiervverfahren (c) Beurteilung der Effizienz der beiden Sortiervverfahren, untere Schranke für die Laufzeit von Sortieralgorithmen	• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	<i>Beispiel:</i> Test- und Analyseumgebung für Sortieralgorithmen Listen mit unterschiedlich vielen Listenelementen und unterschiedlicher Vorsortierung werden bezüglich der Anzahl der Vergleiche von Listenelementen analysiert. Die untere Schranke für die Laufzeit wird bestimmt. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.III - Suchen und Sortieren

Unterrichtsvorhaben Q1-IV:

Thema: Modellierung, Implementierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Nutzung von relationalen Datenbanken (a) Aufbau von Datenbanken und Grundbegriffe - Entwicklung von Fragestellungen zur vorhandenen Datenbank - Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Relationenschema, Datenbankschema (b) SQL-Abfragen - Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle - Analyse und Erarbeitung von SQL-Abfragen auf einer und mehreren Tabelle zur Beantwortung der Fragestellungen (JOIN und Varianten, UNION, AS,	Die Schülerinnen und Schüler - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D), - modifizieren eine Datenbankmodellierung (M), - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - bestimmen Primär- und Sekundärschlüssel (M), - implementieren ein relationales Datenbankschema als Datenbank (I), - analysieren und erläutern eine Datenbankmodellierung (A),	<i>Beispiel:</i> VideoCenter VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontend zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunterzuladen und lokal zu installieren. Unter <i>Beispiel:</i> Schulbuchausleihe Unter www.brd.nrw.de/lerntreffs/informatik/struktured/material/sek2/datenbanken.php (abgerufen: 01. 02. 2016) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.

GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, AVG, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) (c) Vertiefung an einem weiteren Datenbankbeispiel	• erläutern die Eigenschaften normalisierter Datenbankschemata (A), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), •überführen Datenbankschemata in die 1. bis 3. Normalform (M), •ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),	
2. Modellierung von relationalen Datenbanken (a) Entity-Relationship-Diagramm • Ermittlung von Entitätstypen, Attributen zugehöriger Entitäten, Beziehungstypen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms • Erläuterung und Modifizierung einer Datenbankmodellierung (b) Entwicklung einer Datenbank aus einem Datenbankentwurf • Modellierung eines relationalen Datenbankschemas zu einem Entity-	• 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), • erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A).	<i>Beispiel: Fahrradverleih</i> Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kundinnen und Kunden. Die Kundinnen und Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kundinnen und Kunden von <i>BTR</i> können CityBikes, Trekkingräder und Mountainbikes ausleihen. <i>Beispiel: Reederei</i> Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem

Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln (c) Redundanz, Konsistenz und Normalformen • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)		Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird. <i>Beispiel: Buchungssystem</i> In dem Online-Buchungssystem einer Schule können die Lehrenden Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden. Unter http://mrbs.sourceforge.net (abgerufen: 01. 02. 2016) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt geht. <i>Beispiel: Schulverwaltung</i> In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schülerinnen und Schüler, Lehrende und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrerin Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die
--	--	---

		Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und auf das Thema Datenschutz eingegangen werden.
--	--	--

Unterrichtsvorhaben Q1 – V:

Thema: Modellierung und Implementierung von dynamischen nicht-linearen Datenstrukturen und von Anwendungen mit dynamischen nicht-linearen Datenstrukturen in kontextbezogenen Problemstellungen.

Leitfragen: *Wie können Daten im Anwendungskontext mit Hilfe von Graphen und binärer Baumstrukturen verwaltet werden? Wie kann in einem Graphen ein beliebiger, alle oder der kürzeste Weg zwischen zwei Knoten effizient gefunden werden? Welche Kanten müssen in einem zusammenhängenden Graphen mindestens verbleiben, sodass dieser bei minimaler Summe der Kantengewichte weiterhin zusammenhängend ist. Wie kann ich mit einem Binärbaum Daten sortiert verwalten?*

Vorhabenbezogene Konkretisierung:

Anhand von Beispielen für Graphen werden grundlegende Begriffe eingeführt und die beiden Darstellungsformen Adjazenzmatrix und Adjazenzliste erarbeitet. Die Funktionalität der Methoden der Klassen Graph, Vertex und Edge aus den Vorgaben für das Zentralabitur NRW wird erarbeitet. Die Fragestellung nach der Suche eines oder aller Wege zwischen zwei Knoten in einem Graphen motiviert die Erarbeitung von Algorithmen zur Tiefen- und Breitensuche, mit denen Graphen systematisch durchsucht werden können. Anschließend werden anhand von Anwendungskontexten Algorithmen für die Bestimmung kürzester Wege in einem Graphen sowie zur Konstruktion von minimalen Spannbäumen modelliert und zum Teil auch implementiert.

Die Datenstruktur Binärbaum mit den notwendigen Begrifflichkeiten wird als Spezialfall eines Graphen anhand von Anwendungskontexten erarbeitet. Die entsprechende Klasse `BinaryTree<ContentType>` der Vorgaben für das Zentralabitur NRW wird zur Modellierung und Implementierung verschiedener Problemstellungen verwendet. Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und In-Order) implementiert werden.

Mithilfe einer anwendungsorientierten Problemstellung werden die Operationen der Datenstruktur Suchbaum thematisiert und die Klasse `BinarySearchTree<ContentType>` (der Materialien für das Zentralabitur in NRW) modelliert und zumindest partiell implementiert.

Zeitbedarf: 40 Stunden

Sequenzierung des Unterrichtsvorhabens:

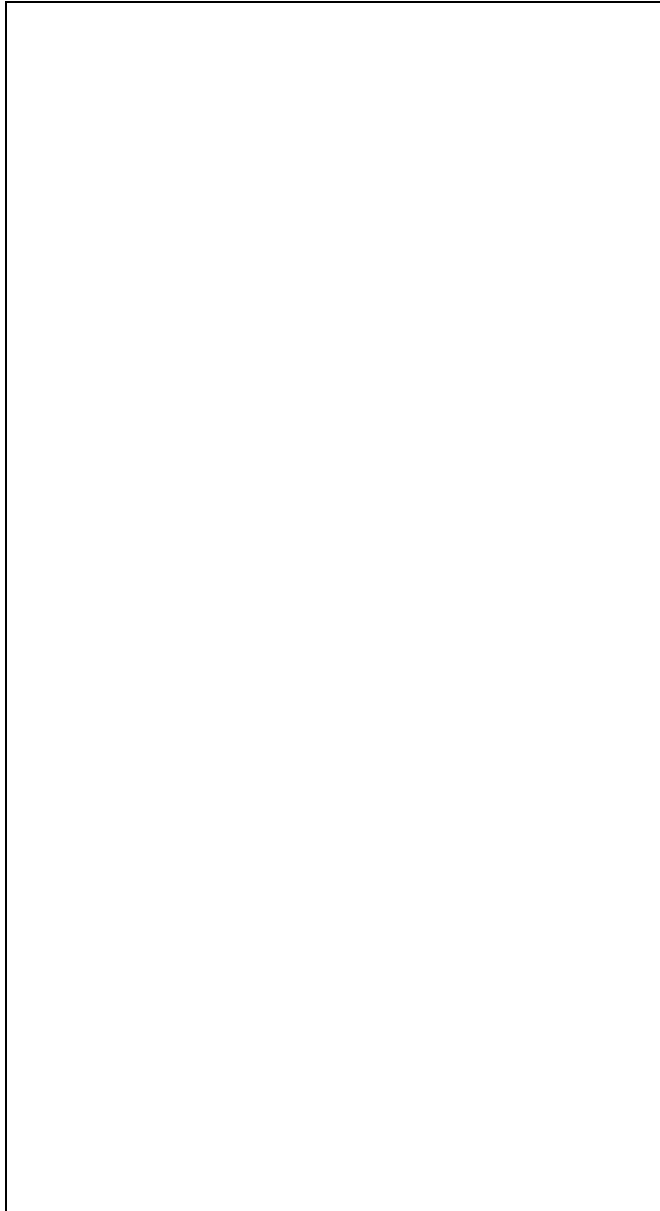
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Analyse von Graphen in verschiedenen Kontexten (a) Grundlegende Begriffe (Graph, gerichtet – ungerichtet, Knoten, Kanten, Kantengewicht) (b) Aufbau und Darstellung von Graphen anhand von Graphenstrukturen in verschiedenen Kontexten (Adjazenzmatrix, Adjazenzliste)	Die Schülerinnen und Schüler • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (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 oder lineare und nichtlineare Datensammlungen zu (M),	<i>Beispiel: „Das Haus vom Nikolaus“</i> Das Haus vom Nikolaus ist das bekannteste Beispiel für einen Graphen, für den ein Eulerweg, aber kein Eulerkreis existiert. Anhand dieses Beispiels werden die Grundbegriffe der Graphentheorie sowie die Darstellung eines Graphen als Adjazenzmatrix eingeführt. <i>Beispiel Soziale Netzwerke</i> Da es in dem Graph eines sozialen Netzwerks im Verhältnis zu den Knoten nur wenige Kanten gibt, bietet sich dieses Beispiel zur Einführung der Darstellung eines Graphen in Form von Adjazenzlisten an. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben LK-Q1.V

	• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“(M), • entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M), • implementieren Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten) (I),	
2. Die Datenstruktur Graph im Anwendungskontext unter Nutzung der Klassen Graph, Vertex und Edge. (a) Erarbeitung der Klassen Graph, Vertex und Edge und beispielhafte Anwendung der Operationen (b) Bestimmung von Wegen in Graphen im Anwendungskontext (Tiefensuche, Breitensuche) (c) Bestimmung von kürzesten Wegen in Graphen im Anwendungskontext (Backtracking, Dijkstra). (d) Bestimmung von minimalen Spannbäumen eines Graphen im Anwendungskontext.		<i>Beispiel: Soziale Netzwerke</i> Ausgehend von dem Problem der Berechnung der Dichte eines sozialen Netzwerkes werden die Funktionalitäten der Methoden der Klassen Graph, Vertex und Edge erarbeitet und erste Beispiele modelliert und implementiert: • Konstruktion eines Beispielgraphen • Anzahl der Knoten • Summe der Kantengewichte • Anzahl der Nachbarn eines Knotens <i>Beispiel: Wegsuche</i> Ausgehend von dem Problem der Suche eines beliebigen Weges zwischen zwei Knoten in einem Graphen wird der Backtracking-Algorithmus zur Tiefensuche erarbeitet. Durch Wegfall der Abbruchbedingung „Zielknoten gefunden“

	• 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 und mithilfe von Testanwendungen(I), • wenden didaktisch orientierte Entwicklungsumgebungen zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),	lassen sich mit dem Algorithmus alle Wege zwischen Start und Zielknoten finden. Als Alternative wird der Algorithmus zur Breitensuche erarbeitet, der als Ergebnis eine Liste aller Knoten, die auf dem Weg vom Start- zum Zielknoten gefunden wurde, zurückgibt. Ausgehend vom Zielknoten kann durch Vorgängersuch in dieser Liste ein Weg vom Start- zum Zielknoten gefunden werden. Damit wird das Verfahren beim Dijkstra-Algorithmus vorbereitet. <i>Beispiel:</i> Kürzeste Wege Ausgehend vom Backtracking-Algorithmus zur Bestimmung aller Wege von einem Start- zu einem Zielknoten in einem Graphen wird ein Algorithmus zur Bestimmung des kürzesten Weges erarbeitet. Aufwandbetrachtungen führen zu der Frage nach einem effizienteren Algorithmus. Der Dijkstra-Algorithmus kann durch geschickte Aufgabenstellung, ggf. unter Einbeziehung des in den Materialien enthaltenen Programms
--	--	--

	• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K).	<i>GraphTool</i> von der Lerngruppe selbstständig erarbeitet und auf mehrere Beispiele angewandt werden. Die Implementierung erfolgt in der Lerngruppe arbeitsteilig unter Vorgabe einer Benutzungsoberfläche. Der Vergleich der beiden Algorithmen unter Effizienzaspekten ist Bestandteil des Unterrichts. <i>Beispiel: Versorgungsnetz</i> Die Problemstellung, Verbraucher eines Dorfes möglichst kostengünstig an ein Versorgungsnetz (Kabel, Gas, Telefon) anzuschließen, motiviert die Behandlung des minimalen Spannbaumes eines Graphen. Die Definition eines Baumes und eines Spannbaumes als Spezialfälle von Graphen bereiten die nächste Unterrichtssequenz vor. <i>Materialien:</i> Ergänzungsmaterialien zum
--	--	--

		Lehrplannavigator Unterrichtsvorhaben LK-Q1.V
3. Die Datenstruktur Binärbaum als Spezialfall eines Graphen im Anwendungskontext unter Nutzung der Klasse <code>BinaryTree<ContentType></code> (a) Definition eines Binärbaums und grundlegende Begriffe (b) Erarbeitung der Klasse <code>BinaryTree<ContentType></code> und beispielhafte Anwendung der Operationen (c) Implementierung der Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf (d) Modellierung und Implementierung einer Anwendung unter Verwendung der Datenstruktur Binärbaum		<i>Beispiel:</i> MorseBaum Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, sodass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Anhand dieses Beispiels werden die Definition eines Binärbaums als Spezialfall eines Graphen und eine rekursive Definition erarbeitet. Die Methoden der generischen Klasse <code>BinaryTree<ContentType></code> eingeführt und zur Implementation des Morsecodierers und -dekodierers genutzt. Wenn man bei der Wurzel startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man den Morsecode des Buchstabens. Im Leistungskurs wird auch ein rekursiver Algorithmus zur Dekodierung von Morsezeichen entwickelt. <i>Beispiel:</i> Termbaum



Arithmetische Ausdrücke werden in einem Binärbaum dargestellt. Mit den Traversierungsalgorithmen lassen sich die Terme in Pre, Post- und In-Order-Darstellung ausgeben. Gegebenenfalls kann ein Interpreter für einen arithmetischen Term mit den vier Grundrechenarten entwickelt werden. Im Unterrichtsvorhaben (Q2-III) kann das Beispiel unter dem Thema „Parsen eines einfachen Terms und die Erzeugung eines Termbaums“ wieder aufgegriffen werden.

Beispiel: Informatikerbaum als Binärbaum
In einem *binären Baum* werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)

Folgende Funktionalitäten werden benötigt:

- Einfügen der Informatiker-Daten in den Baum
- Suchen nach einem Informatiker über den Schlüssel Name

		• Ausgabe des kompletten Datenbestandes in nach Namen sortierter Reihenfolge Anhand des Beispiels werden die Eigenschaften und die Methoden eines binären Suchbaums entwickelt und implementiert. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben LK-Q1.V
4. Erarbeitung, Implementierung und Verwendung der Datenstruktur binärer Suchbaum im Anwendungskontext (a) Erarbeitung der Eigenschaften eines binären Suchbaums im Anwendungskontext (b) Erarbeitung der Attribute und Methoden der generischen Klasse <code>BinarySearchTree<ContentType></code> und des Interfaces <code>ComparableContent</code> (c) Implementierung des Konstruktors und der Methode <code>search</code> der Klasse <code>BinarySearchTree<ContentType></code> (d) Implementierung eines Anwendungsbeispiels einschließlich		<i>Beispiel:</i> Informatikerbaum binärer Suchbaum Das Beispiel wird wieder aufgegriffen und diesmal mit der Klasse <code>BinarySearchTree<ContentType></code> implementiert. Durch Modifikation der implementierten Methoden des Interfaces <code>ComparableContent</code> wird der Suchbaum nach den Geburtsdaten der Informatiker sortiert. <i>Beispiel:</i> Buchindex Als weiteres Anwendungsbeispiel, das mehrere dynamische Datenstrukturen miteinander verknüpft, soll ein Programm modelliert und implementiert werden, das das Stichwortregister eines Buches

der sortierten Ausgabe eines binären
Suchbaumes

verwaltet. Die Wörter werden in einem
binären Suchbaum verwaltet, die
zugehörigen Seitenzahlen als lineare Listen.

Materialien:

Ergänzungsmaterialien zum
Lehrplannavigator Unterrichtsvorhaben
LK-Q1.VI

Unterrichtsvorhaben Q1-VI

Thema: Projektorientierte Softwareentwicklung am Beispiel einer Anwendung mit Datenbankanbindung

Leitfrage: *Wie kann ein komplexeres Java-Projekt unter Verwendung einer Datenbank realisiert werden?*

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt des vorliegenden Unterrichtsvorhabens liegt in der schülerorientierten Erarbeitung eines Java-Projekts, das einer Verknüpfung zwischen der Modellierung und Abfrage von Datenbanken sowie der Entwicklung von Problemlösungen mit Hilfe dynamischer Datenstrukturen wie zum Beispiel der linearen Liste oder dem Graphen herstellt. Dabei soll ein möglichst vollständiger Softwareentwicklungszyklus durchlaufen und sowohl die Arbeit mit Datenbanken als auch mit dynamischen Datenstrukturen vertieft bzw. geübt werden.

Dazu wird zunächst durch die Schülerinnen und Schüler eine lebensweltnahe Problemstellung entwickelt, die sich auch direkt aus den Anforderungen des Schulalltags ergeben und zu einem real einsetzbaren Softwareprodukt führen kann. Dabei könnte es sich zum Beispiel um ein Ausleihsystem für die Schülerbibliothek, eine Datenbank für Fehlstunden oder die Verwaltung von Ergebnissen vom Sportfest handeln. Da es nicht immer einfach ist, eine so praxisorientierte Problemstellung zu finden, sind aber auch andere Projekte denkbar.

Die Wahl der Problemstellung sollte je nach Lerngruppe einen Schwerpunkt auf die Datenbankmodellierung und Abfrage oder aber auf die Arbeit mit dynamischen Datenstrukturen legen. Soll sich das Projekt auf Datenbanken konzentrieren, ist z.B. ein Quizspiel denkbar, das schulweit Fragen aus einer zentralen Datenbank abrufen, Ergebnisse und Ranglisten aller Spielerinnen und Spieler verwaltet und ggf. diese auch gegeneinander antreten lässt, indem ihnen die gleichen Fragen gestellt werden und sie somit in einen direkten Vergleich treten. Will man den Schwerpunkt des Projekts auf dynamische Datenstrukturen legen, wäre anknüpfend an das Unterrichtsvorhaben „Q1-V: Graphen“ die Entwicklung eines Routenplaners, zum Beispiel basierend auf einer Datenbank mit echten Daten des deutschen Autobahnnetzes, ein geeignetes Projekt.

Der Aufbau dieses Projekts orientiert sich an einer vereinfachten Version des Wasserfallmodells der Softwareentwicklung, bestehend aus Analyse, Modellierung, Implementierung und Test (und ggf. auch Installation und Schulung). Insbesondere die Modellierung und Implementierung beziehen sich dabei gegebenenfalls auf die Entwicklung einer geeigneten Datenbank und deren Abfrage und

Manipulation mit SQL und auf die Entwicklung eines entsprechenden Java-Programms, das die Datenbank nutzt. Bei Projekten, die Datenbanken mit einer großen Anzahl von Datensätzen erfordern, sollten die Datensätze in geeigneter elektronischer Form vorgegeben werden. Zur Abfrage und Manipulation der Daten kommen didaktisch vereinfachte Klassen zur Einbindung einer Datenbank in ein Java-Programm zum Einsatz (siehe Abiturklassen zu Datenbanken).

Um den normalerweise hohen Zeitbedarf für Projektarbeiten möglichst gering zu halten, sollte arbeitsteilig vorgegangen werden und auch auf das Prinzip des Prototypings, d.h. die Vervollständigung eines vom Lehrenden vorgegebenen Teilprogramms, zurückgegriffen werden. So sollte zum Beispiel die zeitaufwändige, aber wenig ergiebige Implementation einer grafischen Benutzungsoberfläche – nicht jedoch deren Design – den Schülerinnen und Schülern abgenommen werden.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Analyse einer lebensweltnahen Problemstellung im Hinblick auf die Entwicklung eines Java-Programms mit Datenbankanbindung (a) Entwicklung einer Programmidee (b) Analyse des Problembereichs (c) Entwicklung eines Anforderungskatalogs für das zu entwickelnde Programm	Die Schülerinnen und Schüler • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M),	<i>Beispiel:</i> Quizspiel Verwaltung von Quizfragen, Antworten und Ranglisten <i>Beispiel:</i> Navigationssystem Ermittlung von kürzesten Wegen im deutschen Autobahnnetz <i>Beispiel:</i> Verwaltung der Schülerbücherei Verwaltungsprogramm für das Einpflegen und Ausleihe von Büchern der Schülerbibliothek

	• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • bestimmen Primär- und Sekundärschlüssel (M),	<i>Beispiel:</i> Fehlstundenverwaltung Verwaltungsprogramm für die Fehlstunden von Schülerinnen und Schülern <i>Beispiel:</i> Sportfestverwaltung Verwaltung von Aufgaben, Sportereignissen und Ergebnissen des Schulsportfestes <i>Beispiel:</i> Materialverwaltung Materialien für Vertretungsstunden sollen verwaltet werden.
2. Modellierung einer datenbankgestützten Problemlösung unter Berücksichtigung des MVC-Prinzips (a) Strukturierung nach dem MVC-Prinzip (b) Modellierung der Datenbank (ER-Diagramm, Datenbankschema) (c) Modellierung der Kontrollklassen und Entwicklung von SQL-Anweisungen entsprechend des Anforderungskatalogs		<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Dokumentation der Abiturklassen zur Datenbankbindung Ergänzungsmaterialien zum Lehrplannavigator – Projekt Navigationssystem Ergänzungsmaterialien zum Lehrplannavigator – Projekt Quizspiel

(d) Modellierung einer grafischen Benutzungsoberfläche	• implementieren ein relationales Datenbankschema als Datenbank (I),	
3. Umsetzung des Softwareprojektes (a) Umsetzung der Datenbank in einem Datenbankmanagementsystem (b) Implementierung der Kontrollklassen mit Anbindung an die Datenbank unter Verwendung didaktischer Klassen (c) Integration in den Prototypen der Benutzeroberfläche	• analysieren und erläutern eine Datenbankmodellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), • überführen Datenbankschemata in die 1. bis 3. Normalform (M), • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Dokumentation der Abiturklassen zur Datenbankanbindung Ergänzungsmaterialien zum Lehrplannavigator – Projekt Navigationssystem Ergänzungsmaterialien zum Lehrplannavigator – Projekt Quizspiel
4. Optional: Einführung und Evaluation (a) Installation und Test des Endprodukts im konkreten Anwendungskontext (b) Schulung von Anwendern an der neuen Software (c) Evaluation des Projekts		

	• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • 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),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),	
--	---	--

	• erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A).	
--	--	--

Unterrichtsvorhaben Q2-I:

Thema: Sicherheit und Datenschutz in Informatiksystemen sowie Grenzen und Auswirkungen der Automatisierung

Leitfragen: Welche moralische und rechtliche Verantwortung tragen Informatikerinnen und Informatiker hinsichtlich des Datenschutzes, des Urheberrechts und der gesellschaftlichen Auswirkungen informatischer Systeme? Wie kann Datensicherheit z. B. mit Hilfe von Verschlüsselungsverfahren gewährleistet werden? Wo liegen die Grenzen der Automatisierbarkeit?

Vorhabenbezogene Konkretisierung:

Die Schülerinnen und Schüler werden mit Hilfe eines einprägsamen Fallbeispiels für die Problembereiche des Datenschutzes, des Urheberrechts und der moralischen Verantwortung von Informatikerinnen und Informatikern sensibilisiert. Ein mögliches Fallbeispiel stellt das selbstfahrende Automobil dar, das aus der Perspektive unterschiedlicher Interessensgruppen zu betrachten ist. Im Bereich des Datenschutzes könnte das Erstellen von Bewegungsprofilen thematisiert werden, im Bereich des Urheberrechts, inwieweit Quellcode zur Steuerung des Fahrzeugs geschützt und ggf. sogar nicht einsehbar sein darf, obwohl das Leben von Menschen von seiner fehlerfreien Funktion abhängt. Bezogen auf eine moralische Dimension könnte thematisiert werden, wie ein solches Fahrzeug bei Unfällen reagieren sollte, bei denen ein Personenschaden nicht abzuwenden ist, d.h. z.B. entschieden werden muss, ob ein Ausweichmanöver zur Rettung von Passanten gefahren werden soll, obwohl dabei der Insasse des Fahrzeugs in Lebensgefahr gebracht wird. Auch allgemeine gesellschaftliche Auswirkungen können thematisiert werden, wie z. B. die Konsequenzen für den Arbeitsmarkt, wenn selbstfahrende Autos, LKWs und Züge die Norm werden. Schülerinnen und Schüler sollen dieses Fallbeispiel zunächst hinsichtlich dieser Problemstellungen analysieren und erste Lösungsansätze auf dem Hintergrund ihres Vorwissens erarbeiten.

Anschließend werden die Themen Datenschutz, Urheberrecht und moralische Verantwortung systematisiert und vertieft. Im Bereich Datenschutz werden grundlegende Begriffe (z. B. personenbezogene Daten, informationelle Selbstbestimmung, Datensparsamkeit usw.) eingeführt und an weiteren Fallbeispielen verdeutlicht. Im Bereich Urheberrecht sollte mindestens ein verbreitetes Lizenzsystem thematisiert werden (z. B. Creative-Commons-Lizenzen) und anhand von Beispielen verdeutlicht werden. Im Bereich der moralischen Verantwortung sollte ein Bewertungsmaßstab für moralische Fragen erarbeitet werden (z. B. Grundidee des klassischen

Handlungsutilitarismus nach Jeremy Bentham, Grundidee der Verantwortungsethik nach Hans Jonas). Auch diese Positionen werden auf weitere Beispiele angewendet. Die Erarbeitung der Schwerpunkte Datenschutz, Urheberrecht und moralische Verantwortung kann dabei sequenziell mit der gesamten Lerngruppe oder parallel in zieldifferenten Teilgruppen erfolgen. In beiden Fällen müssen die Ergebnisse zusammengefasst in der gesamten Lerngruppe gesichert werden.

Abschließend wird das Eingangsproblem, in diesem Fall das selbstfahrende Automobil, auf Grundlage der neu erarbeiteten Positionen abschließend bewertet. In diesem Zusammenhang wird insbesondere das Verfassen einer Stellungnahme im Sinne einer reflektierten Darstellung der eigenen Position eingeübt. Eine allgemeingültige und unbestreitbare Bewertung ist aufgrund der Ambivalenz der Beispiele nicht möglich. Der Blick auf die gesamtgesellschaftlichen Konsequenzen selbstfahrender Automobile führt zu der Frage, welche Grenzen der Automatisierung allgemein gesetzt sind.

Nachdem der erste Teil des Unterrichtsvorhabens die Notwendigkeit verdeutlicht hat, insbesondere personenbezogene Daten zu schützen, wird im zweiten Teil des Vorhabens dieses Problem aufgegriffen. Dabei werden die grundlegenden Sicherheitsziele *Vertraulichkeit*, *Integrität* und *Verfügbarkeit* und weitere Sicherheitsziele eingeführt und an Fallbeispielen diskutiert und gegenübergestellt. Anschließend werden symmetrische und asymmetrische Verschlüsselungsverfahren zur Umsetzung von Sicherheitszielen eingeführt. Dabei werden die grundlegenden Verfahren dieser Verschlüsselungen und die Möglichkeiten eines Angriffs auf sie in den Mittelpunkt gestellt. Im Kontext der symmetrischen Verschlüsselungsverfahren sollte das Problem des Schlüsselaustausches angesprochen werden. Im Anschluss an asymmetrische Verschlüsselungsverfahren wird das Prinzip des Signierens thematisiert.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Einführung in die Problemfelder Datenschutz, Urheberrecht und moralische Verantwortung	Die Schülerinnen und Schüler • untersuchen und bewerten anhand von Fallbeispielen	<i>Beispiel: autonomes Fahren</i> Ein namhaftes Softwareunternehmen möchte den Automobilmarkt mit selbstfahrenden Autos

a. Vorstellung eines komplexen Fallbeispiels b. Erarbeitung von Interesse verschiedener Interessensgruppen im Hinblick auf die Problemfelder c. Erster Bewertungsversuch auf Grundlage des Vorwissens von Schülerinnen und Schülern	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). • analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A). • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur	revolutionieren. Der Quellcode zu Steuerung der Fahrzeuge ist Firmengeheimnis, zur Verbesserung der Fahrleistung werden Bewegungsprofile erstellt und noch ist nicht klar, wie das Fahrzeug bei drohenden Unfällen mit Personenschaden reagieren soll. Eine erfolgreiche Einführung der Technologie würde den Straßenverkehr sicherer, schneller und ökologischer machen, allerdings auch zu tausenden von Arbeitslosen durch Wegfall ganzer Berufszweige führen. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I – Fallbeispiel „autonomes Fahren“
2. Erarbeitung grundlegender Positionen (ggf. in zieldifferenten Gruppen) a. Erarbeitung von Grundideen des Datenschutzes (z.B. personenbezogene Daten, informationelle Selbstbestimmung, Datensparsamkeit usw.) b. Erarbeitung von Grundideen des Urheberrechts anhand eines verbreiteten Lizenzsystems c. Erarbeitung eines einfachen moralischen	Internet reflektiert zur	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I – Grundlagen „Informatik, Mensch und Gesellschaft“ (Download LK-Q2-I.2) <i>Beispiel: Creative-Commons-Lizenzen</i> URL: http://de.creativecommons.org (Abgerufen am 19.06.2016)

Bewertungsmaßstabes d. Anwendung auf Fallbeispiele	Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	
3. Zusammenfassung und Sicherung a. Darstellung der relevanten Aspekte zum Datenschutz, Urheberrecht und zur moralischen Bewertung b. Demonstration an jeweils einem einfachen Beispiel <i>Anmerkung:</i> Dieser Schritt ist entscheidend, wenn in die Punkte 2 bis 4 arbeitsteilig umgesetzt wurden.		<i>Beispiel: Stellwände</i> Eine Zusammenfassung und Sicherung kann in Form von Plakatstellwänden erfolgen. Diese können auch im Sinne einer Dauerausstellung im Schulgebäude präsentiert werden.
4. Stellungnahme zu einem komplexen Fallbeispiel a. Reflexion des Einstiegsbeispiels des Unterrichtsvorhabens auf Grundlage der erarbeiteten Positionen zum Datenschutz,		<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I – Materialblatt „Stellungnahme“ (Download LK-Q2-I.3)

Urheberrecht und zur moralischen Bewertung b. Einführung in das Verfassen von Stellungnahmen c. Erarbeitung von Stellungnahmen zum Einstiegsbeispiel d. Diskussion und Würdigung von Stellungnahmen mit Blick auf die Grenzen der Automatisierbarkeit		
5. Sicherheitsziele und der kryptographische Ansatz a. Beschreibung und Gegenüberstellung von Sicherheitszielen anhand von Fallbeispielen b. Problematisierung von Sicherheitszielen in Bezug auf die Kommunikation über offene Kanäle c. Einführung in die Kryptologie – Erreichen von Sicherheitszielen durch Verschlüsselung		<i>Beispiel:</i> Sicherheitsziele <i>Zugriffskontrolle</i> und <i>Authentifikation</i> anhand des „Keyless Entry-System“ bei Autos <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I – Sicherheitsziele und Kommunikationsrollenspiel

6. Verschlüsselungsverfahren und deren Sicherheit a. Monoalphabetische Verschlüsselungsverfahren und Angriffsmöglichkeiten auf diese Verfahren b. Polyalphabetische Verschlüsselungsverfahren und Angriffsmöglichkeiten auf diese Verfahren c. Schlüsseltausch als Grundproblem symmetrischer Verschlüsselungsverfahren (Diffie-Hellman) d. Das Schlüsselpaar und die Einwegfunktion als zentrale Konzepte der asymmetrischen Verschlüsselung e. Probleme der asymmetrischen Verschlüsselung f. Signaturen als Anwendungsgebiet asymmetrischer Kryptographie		<i>Beispiel: Caesar-Verschlüsselung</i> Die Cäsarchiffre als Beispiel für ein monoalphabetisches Verfahren und die Exhaustionsmethode (Brute-Force-Methode) zur Verdeutlichung der Bedeutsamkeit eines starken Schlüssels <i>Beispiel: Vigenère-Verschlüsselung</i> Die Vigenère-Chiffre als Verbesserung einer monoalphabetischen Verschlüsselung und der Kasiski-Test als Grundlage eines verbesserten Angriffsverfahrens <i>Beispiel: One-Time-Pad</i> Der One-Time-Pad als Spezialfall eines sicheren polyalphabetischen Verschlüsselungsverfahrens <i>Beispiel:</i> Das RSA-Verfahren als Beispiel für eine asymmetrische Verschlüsselung mit einer Einwegfunktion <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Verschlüsselungsverfahren <i>Materialien:</i> URL: https://www.cryptool.org/de/ (Abgerufen am 19.06.2016)
--	--	--

Unterrichtsvorhaben Q2-II

Thema: Grundlagen der Netzwerkkommunikation sowie Modellierung und Implementierung von Client-Server-Anwendungen in kontextbezogenen Problemstellungen

Leitfragen: *Wie werden Daten in Netzen übermittelt? Wie entwickelt man ein Client-Server-System im Anwendungskontext? Welche Algorithmen sind zu implementieren?*

Vorhabenbezogene Konkretisierung:

Zunächst werden die Grundlagen von Datenübertragung in Netzwerken erarbeitet. Den Einstieg bildet ein Vergleich der Kommunikation in Netzen mit der physikalischen Zustellung von Sendungen durch Postunternehmen, der zu einem Schichtenmodell als Strukturierungsprinzip für Netzwerkkommunikation führt. Im Anschluss werden von den Schülerinnen und Schülern Antworten auf grundsätzliche Herausforderungen im Bereich Netzwerkkommunikation erarbeitet: die Wahl einer geeigneten Codierung, Vor- und Nachteile verschiedener Topologien, Adressierung/Routing in IP-Netzen sowie die Gestaltung von Protokollen für die Anwendungsebene.

In einer zweiten Phase werden zunächst Clients für vorhandene Server-Dienste entwickelt. Darauf aufbauend können anschließend eigene Server modelliert und implementiert werden.

In einer dritten Phase modellieren und implementieren die die Schülerinnen und Schülern schließlich ein Client-Server-System. Dieses macht u.a. ein Verständnis von Nebenläufigkeit notwendig, da ein Server parallel Nachrichten von mehreren Clients empfangen und verarbeiten können muss.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Grundlagen der Datenübertragung in Netzwerken (a) Schichtenmodell • Erarbeitung eines standardisierten Schichtenmodells für Netzwerkkommunikation (b) Grundlagen der Codierung • Erarbeitung einer eigenen, vereinfachten Codierung für die Bitübertragung (c) Topologien • Erarbeitung und Vergleich ausgewählter Netzwerktopologien (d) Routing • Analyse von Grundlagen der Adressierung in IP-Netzwerken (e) Protokolle • Erarbeitung des beispielhaften Aufbaus eines Protokolls auf der Anwendungsschicht	Die Schülerinnen und Schüler • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Algorithmen und Programme (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), • modifizieren Algorithmen und Programme (I) • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und Beziehungen (M),	<i>Material:</i> Aufgabensammlung Anhand einer Sammlung von Aufgabenblättern (teils inkl. implementierter Begleitwerkzeuge) erarbeiten die Schülerinnen und Schüler im Anwendungskontext Grundlagen der Inhaltspunkte „OSI-Referenzmodell“, „Codierung“, „Topologien“, „Routing“ und „Protokolle“

2. Analyse, Modellierung und Implementierung von Netzwerkanwendungen in Client-Server-Struktur (a) Nutzung einfacher Server-Dienste mittels Clients • Modellierung und Implementierung von Clients für einfach Serverdienste (b) Anbieten von Diensten mittels Server • Analyse vorgegebener Implementationen einfacher Server • Modellierung und Implementierung eigener Server	• modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M), • analysieren und erläutern objektorientierte Modellierungen (A), • stellen Klassen und ihre Beziehungen grafisch dar (D), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).	<i>Beispiel:</i> Echo- bzw. Daytime-Clients und – Server sowie eigene Erweiterungen Anhand der Echo- und Daytime-Dienste (z.B. lokal im Schulnetz durch den Lehrenden zur Verfügung gestellt) erarbeiten die Schülerinnen und Schüler zunächst den die Funktionsweise bzw. den Aufbau einfacher Clients und verwenden dabei zunächst die Klasse Connection, später die Klasse Client. In einem zweiten Schritt implementieren die Schülerinnen und Schüler Daytime- und Echo-Client bzw. Erweiterungen/Abwandlungen derselben (ggf. mit Steigerung des Interaktionsgrades) selbst.
3. Entwicklung eines vollständigen Client-Server-Systems • Protokollentwurf, Dialogorientierung • Modellierung mittels Entwurfs- und Implementationsdiagramm • Bedeutung von Nebenläufigkeit Implementierung		<i>Beispiel:</i> Messenger-Dienst Abschließend entwickeln die Schülerinnen und Schüler ein Client-Server-System zum Versenden von Nachrichten zwischen einzelnen Rechnern (einfacher Messenger), basierend auf selbst gewählten „Nicknames“.

Unterrichtsvorhaben Q2-III:

Thema: Grundlagen von Automaten und formalen Sprachen sowie die Modellierung und Implementierung eines Parsers zu einer formalen Sprache

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

Vorhabenbezogene Konkretisierung:

Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor als Alternative zu einem entsprechenden deterministischen Akzeptor eingeführt. Auch die Umwandlung eines nichtdeterministischen Automaten in einen deterministischen Automaten wird thematisiert.

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

Auch nicht-reguläre Grammatiken werden untersucht, entwickelt oder modifiziert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet. Mit Blick auf diese Einschränkungen endlicher Automaten wird die Idee eines Automaten mit Speicher thematisiert und zu einem Kellerautomaten weiterentwickelt. **Zeitbedarf:** 30 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Endliche Automaten (a) Erarbeitung der formalen Beschreibung eines endlichen Automaten auf der Grundlage von Automaten in bekannten Kontexten (b) Untersuchung, Darstellung und Entwicklung endlicher Automaten (c) Umwandlung nichtdeterministischer endlicher Automaten in deterministische endliche Automaten	Die Schülerinnen und Schüler - analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), - analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A), - erläutern die Grenzen endlicher Automaten und regulärer Sprachen im Anwendungszusammenhang (A), - ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), - entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M), - entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), - entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M),	<i>Beispiele:</i> Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.III
2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen (a) Erarbeitung der formalen Darstellung regulärer Grammatiken (b) Untersuchung, Modifikation und Entwicklung von Grammatiken (c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden	(Fortsetzung der Kompetenzen aus der ersten Sequenz)	<i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliederungsgrammatik <i>Materialien:</i>

(d) Entwicklung regulärer Grammatiken zu endlichen Automaten (e) Entwicklung eines Parsers für eine einfache reguläre Sprache	• modifizieren Grammatiken regulärer und kontextfreier Sprachen (M), • entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D), • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D), entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“, „Teilen	Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.III
3. Grenzen endlicher Automaten		<i>Beispiele:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$ <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.III (Download LK-Q2-III.3)

4. Entwicklung eines Kellerautomaten als Antwort auf die Grenzen endlicher Automaten (a) Erweiterung eines DEA um eine einzelne Speichervariable zum Zählen von Eingabezeichen (z.B. Klammern) und Problematisierung dieses Ansatzes (b) Entwicklung eines Automaten mit Kellerspeicher (c) Anwendung eines Kellerautomaten zur Syntaxüberprüfung auf Grundlage von nicht-regulären Grammatiken (d) Implementierung eines Kellerautomaten zur Syntaxüberprüfung (Backtracking)	und Herrschen“ und „Backtracking“ (M).	<i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.III
--	---	--

Unterrichtsvorhaben Q2-IV:

Thema: Prinzipielle Arbeitsweise eines Computers sowie Modellierung und Implementierung eines Scanners, Parsers und Interpreters für eine einfache maschinennahe Programmiersprache

Leitfragen: *Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinenahen Programms mit diesen Komponenten vorstellen? Wie werden Programme aus höheren Programmiersprachen für den Computer verständlich und wie werden sie in eine tiefere Sprachebene übersetzt und interpretiert?*

Vorhabenbezogene Konkretisierung:

Anhand einer von-Neumann-Architektur und einem maschinennahen Programm wird die prinzipielle Arbeitsweise von Computern verdeutlicht. Grundlegenden Begrifflichkeiten bei der maschinellen Übersetzung von einer Hochsprache in eine maschinenverständliche Sprache werden definiert, veranschaulicht und zum Vorwissen aus dem Unterrichtsvorhaben Q2-III in Beziehung gesetzt.

Ausgehend von einer einfachen formalen Sprache (z. B. eine Konstruktionssprache für geometrische Figuren) werden die Bestandteile eines Compilers dargestellt:

Der Scanner eines Compilers wird in Form eines endlichen Automaten modelliert und implementiert. Die Begriffe *Symboltabelle* und *Tokenliste* werden inhaltlich gefüllt.

Die dem Parser des Compilers zugrunde liegende Grammatik wird in Form einer regulären oder kontextfreien Grammatik definiert und zugehörige Parser-Methoden werden implementiert.

Zum Abschluss wird ein Interpreter-Modul entwickelt, welches die einfache formale Sprache in eine andere Sprachebene übersetzt.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme (a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher (b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann (c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	Die Schülerinnen und Schüler • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • analysieren und erläutern objektorientierte Modellierungen (A),	<i>Beispiel:</i> Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.IV
2. Simulation der Phasen eines Compilers (a) Prozesse beim Compiler: • scannen • parsen • übersetzen/interpretieren (b) Arten von Fehlern: • lexikalischer Fehler • syntaktischer Fehler • semantischer Fehler (c) Einordnung der neuen Begriffe in den Gesamtkontext der formalen Sprachen • Automaten • Grammatiken • Sprachen	(siehe oben)	<i>Beispiel:</i> Scanner, Parser und Interpreter einer Konstruktionssprache für geometrische Figuren Anhand einer einführenden Folienpräsentation werden die Begrifflichkeiten definiert. Ein kleiner Ausschnitt einer Pseudo-Programmiersprache zur Konstruktion von Zeichnungen wird betrachtet. Anfänglich besteht der Sprachumfang aus Programmen mit lediglich einer einzigen Zuweisung. Die grundlegenden Schritte eines Compilers werden am Beispiel dieser Grammatik nachvollzogen. <i>Materialien:</i>

	• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • analysieren und erläutern Algorithmen und Programme (A),	Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.IV
3. Die Schritte eines Compilers (a) Scanner: • endlicher Automat als Grundlage • Vorgabe von Symboltabelle und Tokenliste zur Verwaltung und Erkennung des Quelltextes • Erweiterung des terminalen Alphabets der zu übersetzenden formalen Sprache • Implementierung als endlicher Automat (b) Parser: • reguläre (oder wahlweise kontextfreie) Grammatik als Grundlage • Vorgabe einer Grundversion des Parsers • Erweiterung des Sprachumfangs • Implementierung der Parsermethoden für die Produktionsregeln der kontextfreien Grammatik (c) Interpreter • Vorgabe einer Grundversion des Interpreters • Erweiterung des Sprachumfangs • Implementierung	• modifizieren Algorithmen und Programme (I), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I), • analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A), • ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M), • analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A),	<i>Beispiel:</i> Der Scanner-Automat zur Erkennung der einzelnen Symbole wird als endlicher Automat realisiert. Mithilfe der Symboltabelle wird die Vereinfachung des Automaten deutlich gemacht und der vereinfachte Scanner-Automat wird schrittweise erweitert und implementiert. In der zweiten Phase wird die der Pseudoprogrammiersprache zugrunde liegende Grammatik analysiert. Die Überprüfung der syntaktischen Korrektheit wird in Form eines Parsers modelliert und implementiert. Dabei wird der Sprachumfang der Pseudo-Programmiersprache schrittweise erweitert. In der dritten Phase wird ein zu Teilen bereits vorbereiteter Interpreter analysiert und erweitert, welcher Programme der Pseudo-Programmiersprache zur Konstruktion von Zeichnungen in eine Grafik übersetzt. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.IV (Download LK-Q2-IV.3)

	• modifizieren Grammatiken regulärer und kontextfreier Sprachen (M), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M), • modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I), • nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).	
--	--	--

Unterrichtsvorhaben Q2-V:

Thema: Entwicklung eines Netzwerkspiels mit Durchführung eines vollständigen Softwareentwicklungszyklus

Leitfragen

Wie ist ein Softwareentwicklungszyklus aufgebaut? Welches Protokoll ist für die vorgegebene Funktionalität angemessen bzw. garantiert eine fehlerfreie Kommunikation? Welche Reaktionen auf Kommunikationsereignisse sind server- und clientseitig zu entwickeln? Wie können verwendete Daten mit Hilfe von Graphen verwaltet werden?

Vorhabenbezogene

Konkretisierung:

Anwendungskontext ist ein von den Schülerinnen und Schülern zu entwickelndes Zweipersonen-Netzwerkspiel. Anhand des Spiels sollen die grundlegenden Prinzipien und Begrifflichkeiten der Kommunikation in Netzwerken, dem Aufbau einer Client-Server-Struktur und der Datenorganisation mit Hilfe von Graphen aus den Unterrichtsvorhaben Q1-V und Q2-II festigend wiederholt werden. Grundlage für das Softwareprojekt sind die Klassen zur Netzwerkkommunikation Connection, Client und Server. Der Begriff des Softwareentwicklungszyklus wird thematisiert und es werden alle Phasen der Entwicklung eines Softwareprojekts über Analyse, Modellierung und Implementierung anhand des Netzwerkspiels vollzogen. Das Projekt läuft in folgenden groben Phasen ab:

1. Projekteinstieg / -planung (Spielauswahl, Zeitmanagement)
2. Projektumsetzung (informatische Analyse, Modellierung, Implementierung, Test è ggf. in Zyklen)
 - 2.1 Analyse des Spiels mit dem Ziel einer späteren informatischen Umsetzung als Netzwerkspiel
 - 2.2 Modellierung und Implementierung des Spiels als Netzwerkanwendung
 - 2.3 Reflexion des Softwareprodukts
3. Projektreflexion (Reflexion der Projektplanung, Präsentation)

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Projekteinstieg / -planung (Spielauswahl, Zeitmanagement) 2. Projektumsetzung (informatische Analyse, Modellierung, Implementierung, Test & ggf. in Zyklen) 2.1. <i>Analyse des Spiels mit dem Ziel einer späteren informatischen Umsetzung als Netzwerkspiel</i> (a) Spielen des Spiels in mehreren Kleingruppen in Form eines Rollenspiels nach vereinbarten Regeln: i. Ein Schüler oder eine Schülerin übernehmen die Rolle des Spielleiters oder der Spielleiterin. ii. Ein weiteres Gruppenmitglied protokolliert präzise die Kommunikation zwischen der Spielleiterin bzw. dem Spielleiter und den Spielerinnen und Spielern. (b) Formalisierung des Spielablaufs 2.2. <i>Modellierung und Implementierung des Spiels als Netzerkennung</i> (a) Entwurf eines Protokolls zur Kommunikation zwischen Spielserver	Die Schülerinnen und Schüler - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - analysieren und erläutern objektorientierte Modellierungen (A), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“ (M),	<i>Beispiel:</i> GraphColoringGame Zwei-Personen-Spiel, bei dem die beiden Personen abwechselnd am Zug sind. Das Spielfeld ist ein ungerichteter Graph ohne Mehrfachkanten. Die Knoten des Graphen können mit vorgegebenen Farben markiert werden. Die beiden Spielerinnen bzw. Spieler färben abwechselnd einen noch ungefärbten Knoten. Dabei muss beachtet werden, dass kein adjazenter Knoten mit derselben Farbe markiert wurde. Es verliert derjenige Spielerinn bzw. derjenige Spieler, die bzw. der unter diesen Bedingungen keinen Zug mehr machen kann. Sind alle Knoten korrekt gefärbt, endet das Spiel unentschieden. <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2-V (LK)

und -client basierend auf den erarbeiteten Spielregeln (b) Entwicklung von Entwurfs- und Implementationsdiagrammen für den Spielservers (c) Implementation und Test der Spielservers-Klasse und von dieser benötigter Fachklassen (d) Modellierung, Implementierung und Test des Spielclients (e) Test der Zusammenarbeit von Spielservers und Spielclient 2.3. <i>Reflexion des Softwareprodukts</i> (a) Identifikation der Stufen eines Softwareentwicklungszyklus (b) Wiederholende Darstellung der Entwicklungsschritte zum fertigen Produkt (c) Mögliche Erweiterungen: - ggf. Implementierung einer grafischen Benutzeroberfläche (GUI) für den Client - ggf. Analyse, Modellierung und Implementierung alternativer Spielregeln - ggf. Implementierung einer KI für einen Computer-Gegenspieler	• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I). • analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A), • entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I). • 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).	
---	---	--

3. Projektreflexion (Reflexion der Projektplanung, Präsentation)		
---	--	--

2.4 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

Die Fachkonferenz Informatik des St.-Bernhard-Gymnasiums hat die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen.

Überfachliche Grundsätze:

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

Fachliche Grundsätze:

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

2.5 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

2.5.1 Sekundarstufe I

I) Beurteilungsbereich „Sonstige Mitarbeit“

Die Leistungsbewertung im Beurteilungsbereich „Sonstige Mitarbeit“ bezieht sich auf alle im Zusammenhang mit dem Unterricht erbrachten mündlichen, schriftlichen und praktischen Leistungen. Grundlage sind die im Kernlehrplan Informatik ausgewiesenen Kompetenzen. Die Bewertung erfolgt kontinuierlich sowie kriterienorientiert und berücksichtigt sowohl die Qualität als auch die Kontinuität der Leistungen.

Bestandteile der Sonstigen Mitarbeit

Zur Sonstigen Mitarbeit zählen insbesondere:

- Beiträge zum Unterrichtsgespräch
- Mitarbeit in Partner- und Gruppenarbeitsphasen
- Präsentationen und Kurzvorträge
- Praktische Arbeiten am Computer
- Entwicklung, Implementierung und Test von Programmen
- Dokumentationen und Protokolle
- Arbeitsmappen, Portfolios oder Lerntagebücher
- Ergebnisse projektorientierter Arbeitsformen
- Kurze schriftliche Übungen
- Recherche- und Analyseaufgaben
- Präsentation von Arbeitsergebnissen

Kriterien der Bewertung

A. Mündliche Mitarbeit

Bewertet werden insbesondere:

- Qualität und Fachlichkeit der Beiträge
- Verwendung informatischer Fachsprache
- Problemlösefähigkeit und Transferleistungen
- Selbstständigkeit der Beiträge
- Kontinuität der Mitarbeit
- Fähigkeit zur Reflexion und Begründung
- Eingehen auf Beiträge anderer Schülerinnen und Schüler

B. Praktische Leistungen am Computer / anderen Informatiksystemen

Bewertet werden insbesondere:

- sachgerechter Umgang mit Hard- und Software
- Entwicklung funktionaler Lösungen
- Strukturierung und Dokumentation von Programmen
- Anwendung informatischer Konzepte und Algorithmen
- Fehlersuche und Optimierung
- Selbstständigkeit und Sorgfalt beim Arbeiten
- Einhaltung von Arbeitsaufträgen und Zeitvorgaben

C. Kooperation und Projektarbeit

Bewertet werden insbesondere:

- konstruktive Mitarbeit in Gruppenprozessen
- Zuverlässigkeit und Verantwortungsübernahme
- Kommunikationsfähigkeit
- Planung und Organisation von Arbeitsprozessen
- Präsentation gemeinsamer Ergebnisse
- Reflexion des Arbeitsprozesses

D. Schriftliche Beiträge

Bewertet werden insbesondere:

- fachliche Richtigkeit
- Vollständigkeit
- Strukturierung und Darstellung
- angemessene Verwendung von Fachbegriffen
- Nachvollziehbarkeit der Lösungswege
- Sorgfalt und Übersichtlichkeit

Grundsätze der Leistungsbewertung

- Die Bewertung erfolgt transparent und für die Schülerinnen und Schüler nachvollziehbar.
- Die Kriterien werden zu Beginn des Schuljahres bzw. Unterrichtsvorhabens erläutert.
- Die Leistungsbewertung orientiert sich am individuellen Lernfortschritt sowie an den im Kernlehrplan ausgewiesenen Kompetenzen.
- Neben Ergebnissen werden auch Lernprozesse berücksichtigt.
- Kooperative Arbeitsformen werden angemessen in die Bewertung einbezogen.

Bewertungsmatrix Sonstige Mitarbeit (S I)

Bewertungsbereich	Sehr gut (1)	Gut (2)	Befriedigend (3)	Ausreichend (4)	Mangelhaft / Ungenügend (5–6)
Mündliche Mitarbeit	sehr regelmäßige, eigenständige und fachlich fundierte Beiträge; sicherer Gebrauch der Fachsprache	regelmäßige und überwiegend richtige Beiträge; Fachsprache meist sicher	gelegentliche Mitarbeit; Beiträge teilweise fachlich richtig	seltene Mitarbeit; einfache oder lückenhafte Beiträge	kaum oder keine Mitarbeit; Beiträge überwiegend fehlerhaft
Praktische Arbeit am Computer	entwickelt selbstständig funktionale und strukturierte Lösungen; dokumentiert sicher	arbeitet weitgehend selbstständig; Lösungen überwiegend korrekt	einfache Aufgaben mit Unterstützung lösbar	benötigt häufig Hilfe; Ergebnisse teilweise unvollständig	Aufgaben können kaum bearbeitet werden
Programmierung / Algorithmik	Programme logisch aufgebaut, fehlerarm und gut dokumentiert	Programme überwiegend korrekt und nachvollziehbar	grundlegende Strukturen vorhanden	deutliche Probleme bei Umsetzung und Struktur	keine funktionsfähigen Lösungen
Problemlösen / Transfer	überträgt Kenntnisse sicher auf neue Situationen	erkennt Lösungswege meist selbstständig	löst bekannte Problemtypen	benötigt starke Anleitung	keine eigenständige Problemlösung

Kooperation Gruppenarbeit	/	arbeitet sehr konstruktiv und verantwortungsvoll	kooperiert zuverlässig	beteiligt sich angemessen	Mitarbeit eingeschränkt	verweigert Zusammenarbeit häufig
Präsentation Dokumentation	/	sehr strukturierte, fachsprachlich präzise Darstellung	überwiegend klare Darstellung	verständliche Grundstruktur	unübersichtlich oder lückenhaft	unvollständig oder kaum nachvollziehbar
Arbeitsverhalten		sehr selbstständig, sorgfältig und zuverlässig	überwiegend zuverlässig	meist angemessen	häufig unselbstständig	unzuverlässig und wenig arbeitsbereit

II) Beurteilungsbereich Kursarbeiten (nur WP II)

In der Sekundarstufe I werden im Fach Informatik pro Halbjahr zwei Kursarbeiten geschrieben. Die Termine werden zu Beginn eines jeden Halbjahres von der Mittelstufenkoordination festgelegt. Die Bearbeitungszeit beträgt in der Regel zwischen 60 und 90 Minuten und orientiert sich an den Vorgaben des Kernlehrplans sowie am Alter und Lernstand der Schülerinnen und Schüler.

Die Klassenarbeiten überprüfen die im Unterricht erworbenen informatischen Kompetenzen. Dazu zählen insbesondere Kenntnisse und Fähigkeiten aus den Bereichen Algorithmik und Programmierung, Daten und Codierung, Informatiksysteme sowie Informatik, Mensch und Gesellschaft. Neben der Reproduktion von Fachwissen werden auch die Anwendung informatischer Methoden, Problemlösefähigkeit, Transferleistungen sowie die Analyse und Reflexion informatischer Sachverhalte berücksichtigt.

Die Aufgabenformate können sowohl theoretische als auch praktische Anteile umfassen und beinhalten beispielsweise:

- Analyse und Entwicklung von Algorithmen,
- Verständnis und Erstellung von Programm
- Modellierungsaufgaben,
- Anwendungs- und Transferaufgaben,
- Analyse informatischer Systeme,
- strukturierte schriftliche Darstellungen informatischer Zusammenhänge.

Die Bewertung erfolgt kriterienorientiert und transparent. Berücksichtigt werden fachliche Richtigkeit, Nachvollziehbarkeit der Lösungswege, angemessene Verwendung der Fachsprache sowie die Strukturierung und Darstellung der Ergebnisse.

Grundlage der Leistungsbewertung sind die Vorgaben des Kernlehrplans Informatik NRW sowie die Bestimmungen der APO-S I.

Für die Bewertung der Arbeiten gilt folgende Notentabelle als Orientierung, die sich an die Vorgaben für die Zentralen Prüfungen zum Ende des 10. Schuljahres anlehnt. Die Lehrkraft kann aber im eigenen pädagogischen Ermessen von dieser Zuordnung abweichen.

Prozent der erreichbaren Punkte	Note
---------------------------------	------

95–100 %	1+
----------	----

90–94 %	1
---------	---

85–89 %	1–
---------	----

80–84 %	2+
---------	----

75–79 %	2
---------	---

70–74 %	2–
65–69 %	3+
60–64 %	3
55–59 %	3–
50–54 %	4+
45–49 %	4
40–44 %	4–
33–39 %	5+
27–32 %	5
20–26 %	5–
0–19 %	6

2.5.2 Sekundarstufe II

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

I) Beurteilungsbereich Klausuren

Verbindliche Absprachen:

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

Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr
Dauer der Klausur: 2 Unterrichtsstunden (90 Min)
- Grundkurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 2.1 eine Facharbeit geschrieben werden.

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

Kriterien

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

Die Zuordnung der Hilfspunktsumme zu den Notenstufen orientiert sich in der Einführungsphase an der zentralen Klausur und in der Qualifikationsphase am Zuordnungsschema des Zentralabiturs. Die Note ausreichend soll bei Erreichen von ca. 50% der Hilfspunkte erteilt werden. Von den genannten Zuordnungsschemata kann im Einzelfall begründet abgewichen werden, wenn sich z. B. besonders originelle Teillösungen

nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOST §13 (2)) angemessen erscheint.

II) Beurteilungsbereich Sonstige Mitarbeit

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn eines jeden Schulhalbjahres erläutert.

Leistungsaspekte

Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit und kooperative Kompetenzen in Partner-/Gruppenarbeitsphasen
- Arbeitsverhalten während Erarbeitungsphasen (u. a. Zielstrebigkeit; Trial- & Error-Kompetenz; Ausdauer; Fähigkeit, gezielt Hilfe durch Mitschüler, Lehrkraft oder andere geeignete Quellen in Anspruch zu nehmen; Fähigkeit, Mitschüler zu unterstützen)

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen

- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,

-
- die Präzision und
 - die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

Bewertungsmatrix Sonstige Mitarbeit (SII)

Kompetenzbereich	Kriterien		sehr gut (13–15 P.)	gut (10–12 P.)	befriedigend (7–9 P.)	ausreichend (4–6 P.)	mangelhaft/ungenügend (0–3 P.)
Mündliche Mitarbeit	Qualität der Beiträge	der	analytisch, differenziert, fachlich präzise, weiterführend	fachlich sicher und selbstständig	überwiegend korrekt	eher reproduktiv, teilweise unklar	kaum fachlich verwertbar
	Reflexion und Transfer	und	selbstständige Transferleistungen und Reflexion	angemessene Reflexion	einfache Transferleistungen	nur mit Hilfen	keine Transferfähigkeit
	Fachsprache		sehr sicher und präzise	überwiegend sicher	meist korrekt	teilweise unsicher	kaum angemessen
Algorithmik und Programmierung	Entwicklung von Lösungen	von	kreative, effiziente und strukturierte Lösungen	sichere Lösungsstrategien	funktionale Standardlösungen	einfache Lösungen mit Hilfen	keine tragfähigen Lösungen
	Codequalität		effizient, strukturiert, dokumentiert	weitgehend sauber strukturiert	überwiegend nachvollziehbar	teilweise fehlerhaft	nicht funktionsfähig
	Fehlersuche und Debugging	/	systematisch und selbstständig	überwiegend selbstständig	mit Unterstützung	nur eingeschränkt möglich	kaum möglich
Modellierung und Analyse	Abstraktion und Modellbildung	und	sehr sichere Modellierung komplexer Systeme	sichere Anwendung	grundlegende Modellierung	lückenhafte Modellierung	erhebliche Defizite

		Analyse informatischer Systeme	differenziert reflektiert	und	fachlich korrekt	teilweise nachvollziehbar	oberflächlich	kaum verständlich
Projekt-Teamarbeit	und	Selbstständigkeit	vollständig eigenständig		überwiegend eigenständig	teilweise angeleitet	stark angeleitet	keine selbstständige Arbeit
		Kooperation	konstruktiv, initiativ		zuverlässig	überwiegend kooperativ	eingeschränkt kooperativ	verweigert Mitarbeit
		Präsentation	adressatenorientiert, souverän		strukturiert und verständlich	nachvollziehbar	teilweise unklar	unzureichend
Schriftliche Beiträge / Dokumentation		Fachliche Qualität	präzise differenziert	und	fachlich sicher	überwiegend korrekt	teilweise fehlerhaft	erhebliche Mängel
		Struktur und Darstellung	sehr klar fachgerecht	und	gut strukturiert	nachvollziehbar	teilweise unübersichtlich	nicht nachvollziehbar

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

Zusammenarbeit mit anderen Fächern

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Dies trifft insbesondere auf das Fach Mathematik und ferner auch Physik zu. 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. Im Bereich *Informatik und Gesellschaft* kommen auch fachübergreifende Aspekte mit Gesellschaftswissenschaften in Frage, zum Beispiel mit Sozialwissenschaften oder auch Religionslehre bzw. Philosophie (ethische Aspekte).

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung.

Erstellung der Facharbeit

Für die Facharbeit ist mindestens ein Fachbuch als Quelle zu verwenden und anzugeben. Alle entstandenen Implementationen sind sowohl gedruckt als auch in digitaler Form an die Facharbeit anzuhängen. Sollten Befehle aus Bibliotheken verwendet werden, so sind diese ebenfalls im Anhang zu dokumentieren.

4 Qualitätssicherung und Evaluation

Das schulinterne Curriculum beider Sekundarstufen wird kontinuierlich evaluiert und weiterentwickelt. Dies geschieht insbesondere in den Sitzungen der Fachkonferenzen in Zusammenarbeit mit den Schülerinnen und Schülern, Eltern und den Lehrkräften.