

St.-Antonius-Str.17 51429 Bergisch Gladbach
Tel.: 02204/984503 Telefax: 02204/984530
E-Mail: post@gymnasium-herkenrath.de
Internet: www.gymnasium-herkenrath.de



Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe Informatik – Qualifikationsphase

Stand: 11/2018

Inhalt

1	Rahmenbedingungen des Gymnasiums Herkenrath	3
2	Entscheidungen zum Unterricht	4
2.1	Verwendete Lehrwerke	4
2.2	Vorbemerkungen	4
2.3	Übersichtsraster der Unterrichtsvorhaben	5
3	Konkretisierte Unterrichtsvorhaben	7
3.1	Unterrichtsvorhaben I – WDH Objektorientierte Programmierung:	7
3.2	Unterrichtsvorhaben II – Lineare, dynamische Datenstrukturen	10
3.3	Unterrichtsvorhaben III – Suchen und Sortieren	12
3.4	Unterrichtsvorhaben IV – Relationale Datenbanken	14
3.5	Unterrichtsvorhaben V-GK – Sicherheit und Datenschutz in Netzstrukturen	16
3.6	Unterrichtsvorhaben V-LK – Softwareentwicklung mit Datenbankbindung	17
3.7	Unterrichtsvorhaben VI – Dynamische, nichtlineare Datenstrukturen	20
3.8	Unterrichtsvorhaben VII – Automaten und formale Sprachen	23
3.9	Unterrichtsvorhaben VIII-GK – Grenzen der Automatisierbarkeit	26
3.10	Unterrichtsvorhaben VIII-LK – Arbeitsweise eines Computers	27
3.11	Unterrichtsvorhaben IX-LK – Grenzen und Auswirkung der Automatisierbarkeit	30
3.12	Unterrichtsvorhaben X-LK – Netzwerkkommunikation	34

1 Rahmenbedingungen des Gymnasiums Herkenrath

Das Fach Informatik wird am Gymnasium Herkenrath ab der Jahrgangsstufe 8 im Wahlpflichtbereich dreistündig unterrichtet.

Im Rahmen der alljährlich stattfindenden Methodentage erwerben die Schülerinnen und Schüler der Jahrgangsstufen 5 und 6 einen sogenannten „Computerführerschein“ zum Umgang mit informatischen Systemen der Schule und zur Aneignung von Office-Kenntnissen. Der Kurs ist jedoch nicht unmittelbar dem Fach Informatik zugeordnet.

In der Sekundarstufe II bietet das Gymnasium Herkenrath den Schülerinnen und Schüler in allen Jahrgangsstufen mindestens einen Grundkurs in Informatik an. Zukünftig sollen auch Leistungskurse in Informatik angeboten werden.

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

Das Fach wird im Grundkurs dreistündig unterrichtet, dabei sieht die Blockung eine Doppel- und eine Einzelstunde vor. Der Leistungskurs würde mit zwei Doppelstunden und einer Einzelstunde fünfstündig unterrichtet. Die Grundsätze zur Leistungsbewertung im Fach Informatik sind in einem separaten Dokument, dem Leistungskonzept, beschrieben. Es ist über die Schulhomepage¹ abrufbar.

Die Schule ist mit drei Computerräumen mit je 30 Arbeitsplätzen ausgestattet. Alle Arbeitsplätze sind an das Schulnetz angeschlossen, so dass Schülerinnen und Schüler über einen individuellen Zugang zum zentralen Server der Schule von jedem Arbeitsplatz aus Zugriff auf ihre eigenen Daten haben und den Computer zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können. Die Regeln zur Nutzung der Computerräume sind auf der Schulhomepage veröffentlicht.

Am Gymnasium Herkenrath unterrichten vier Informatiklehrerinnen und -lehrer.

¹ <https://www.gymnasium-herkenrath.de/fach/informatik>

2 Entscheidungen zum Unterricht

2.1 Verwendete Lehrwerke

Gemäß Beschluss der Fachkonferenz werden im Informatikunterricht in der Sekundarstufe II die folgenden Lehrwerke eingesetzt:

- Einführungsphase: Informatik - Lehrwerk für die gymnasiale Oberstufe - Neubearbeitung: Schülerband 1, Schöningh Verlag im Westermann Schulbuch, 2014, ISBN-13: 978-3140371261
- Qualifikationsphase: Informatik - Lehrwerk für die gymnasiale Oberstufe - Neubearbeitung: Schülerband 2, Schöningh Verlag im Westermann Schulbuch, 2015, ISBN-13: 978-3140371278

2.2 Vorbemerkungen

Der schulinterne Lehrplan ist Grundlage sowohl für Leistungskurse wie auch Grundkurse des Fachs Informatik in der Qualifikationsstufe (Q1 und Q2). Kompetenzen, die nur im Leistungskurs verpflichtend vermittelt werden müssen, sind entsprechend markiert.

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

Im „Übersichtsraster der Unterrichtsvorhaben“ wird eine verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Exkursionen o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

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

Während der Fachkonferenzbeschluss zum „Übersichtsraster der Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft hat, beinhaltet die Ausweisung „Konkretisierte Unterrichtsvorhaben“ Beispiele und Materialien, die empfehlenden Charakter haben.

Nach dem ersten durchgeführten LK-Durchgang werden die Unterrichtsvorhaben und deren Verteilung durch die Fachkonferenz evaluiert und gegebenenfalls modifiziert.

2.3 Übersichtsraster der Unterrichtsvorhaben

<p><u>Unterrichtsvorhaben I (S.7):</u> (Q1.1)</p> <p>Thema: <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung (LK: unter Berücksichtigung der Gestaltung einer Benutzungsoberfläche)</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte, Klassen und ihre Beziehungen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: GK: 8 Std. – LK: 15 Std.</p>	<p><u>Unterrichtsvorhaben II (S.10):</u> (Q1.1)</p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen (LK: sowie Modellierung und Implementierung der Datenstrukturen)</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: GK: 20 Std. – LK: 25 Std.</p>	<p><u>Unterrichtsvorhaben III (S.12):</u> (Q1.1)</p> <p>Thema: <i>Suchen und Sortieren auf linearen Datenstrukturen (LK: sowie Analyse und Beurteilung von Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen in kontextbezogenen Problemstellungen)</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: GK: 16 Std. – LK: 20 Std.</p>
<p><u>Unterrichtsvorhaben IV (S.14):</u> (Q1.2 / Q2.1)</p> <p>Thema: <i>Modellierung und Nutzung (LK: sowie Implementierung) von relationalen Datenbanken in Anwendungskontexten</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit <p>Zeitbedarf: GK: 20 Std. – LK: 20 Std.</p>	<p>GK²: <u>Unterrichtsvorhaben V-GK (S.16):</u> (Q1.2)</p> <p>Thema: <i>Sicherheit und Datenschutz in Netzstrukturen</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: GK: 10 Std.</p>	<p>LK³: <u>Unterrichtsvorhaben V-LK (S.17):</u> (Q1.2 / Q2.1)</p> <p>Thema: <i>Projektorientierte Softwareentwicklung am Beispiel einer Anwendung mit Datenbankanbindung</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit <p>Zeitbedarf: LK: 15 Std.</p>

² GK – Kompetenzen und Inhalte ausschließlich für Grundkurse

³ LK – Kompetenzen und Inhalte ausschließlich für Leistungskurse

<p><u>Unterrichtsvorhaben VI (S.20):</u> (LK: Q1.2 / GK: Q2.2)</p> <p>Thema: <i>Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen (LK: sowie Modellierung und Implementierung der Datenstrukturen)</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: GK: 24 Std. – LK: 40 Std.</p>	<p><u>Unterrichtsvorhaben VII (S.23)</u> (Q2.1 / Q1.2)</p> <p>Thema: <i>Automaten und formale Sprachen</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Endliche Automaten (<i>LK: und Kellerautomaten</i>) • Grammatiken regulärer (<i>LK: und kontextfreier</i>) Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: GK: 20 Std. – LK: 30 Std.</p>	<p><u>GK Unterrichtsvorhaben VIII-GK (S.26):</u> (Q2.1)</p> <p>Thema: <i>Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung <p>Zeitbedarf: GK: 12 Std.</p>
<p><u>LK: Unterrichtsvorhaben VIII-LK (S.27):</u> (Q2.1/Q1.2)</p> <p>Thema: <i>Prinzipielle Arbeitsweise eines Computers sowie Modellierung und Implementierung eines Scanners, Parsers und Interpreters für eine einfache maschinennahe Programmiersprache</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementierung von Algorithmen • Scanner, Parser und Interpreter für eine reguläre Sprache • Einzelrechner und Rechnernetzwerke <p>Zeitbedarf: LK: 12 Std.</p>	<p><u>LK: Unterrichtsvorhaben IX-LK (S.30):</u> (Q2.2)</p> <p>Thema: <i>Sicherheit und Datenschutz in Informatiksystemen sowie Grenzen und Auswirkungen der Automatisierung</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Nutzung von Informatiksystemen • Sicherheit • Wirkungen der Automatisierung • Grenzen der Automatisierung <p>Zeitbedarf: LK: 20 Std.</p>	<p><u>LK: Unterrichtsvorhaben X-LK(S. 34):</u> (Q2.2)</p> <p>Thema: <i>Grundlagen der Netzwirkkommunikation sowie Modellierung und Implementierung von Client-Server-Anwendungen in kontextbezogenen Problemstellungen</i></p> <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Algorithmen in ausgewählten Kontexten • Einzelrechner und Rechnernetzwerke • Nutzung von Informatiksystemen <p>Zeitbedarf: LK: 20 Std.</p>

3 Konkretisierte Unterrichtsvorhaben

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren (K)* 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).

Es werden folgende Abkürzungen für die Kompetenzen verwendet: Argumentieren (A), Modellieren (M), Implementieren (I), Darstellen und Interpretieren (D)

3.1 Unterrichtsvorhaben I – WDH Objektorientierte Programmierung:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung (*LK: unter Berücksichtigung der Gestaltung einer Benutzungsoberfläche*)

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.

LK: 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 Benutzungsoberfläche wird von den Schülerinnen und Schüler nach softwareergonomischen Kriterien konzipiert und mit einem GUI-Builder umgesetzt.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(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 (g) Testen einzelner Klassen der Anwendung</p> <p>LK: Entwurf einer grafischen Benutzungsoberfläche</p> <p>(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</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • 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). 	<p><i>Beispiel 1:</i> Helden-und-Monster-Spiel Projekt-einstieg aus dem Lehrwerk „Informatik 2“ (S. 8ff).</p> <p><i>Beispiel 2:</i> Rollerprojekt aus dem Lehrwerk „Informatik 1“ (S.98ff).</p> <p><i>Beispiel LK:</i> Softwareprojekt aus dem Lehrwerk „Informatik 1“ (S.178ff).</p>

<i>einem GUI-Builder</i>	<i>LK:</i> <ul style="list-style-type: none">• <i>entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M),</i>	
--------------------------	--	--

3.2 Unterrichtsvorhaben II – Lineare, dynamische Datenstrukturen

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen (*LK: sowie Modellierung und Implementierung der Datenstrukturen*)

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 (*LK: und implementiert. Es werden unterschiedliche Implementationsmöglichkeiten verglichen.*) Anschließend wird die Klasse Queue vorgestellt. Die für die Problemstellung notwendige Klassen werden modelliert und implementiert, dabei wird die Klasse Queue verwendet.

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.

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

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><i>LK: Modellierung und Implementation einer Datenstruktur Schlange, die die Daten nach dem FIFO-Prinzip im Anwendungskontext verwaltet.</i></p> <p>Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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 	<p><i>Beispiel:</i> Warteschlange in Arztpraxis oder vor Heiler im Heldenspiel</p> <p>Die Verwaltung der Warteschlange erfolgt über eine Klasse. Wesentliche Operationen sind das „Hinzufügen“ einer Person und das „Entfernen“ dieser, wenn ihr Anliegen bearbeitet wurde.</p>

mehrerer Objekte der Klasse Queue	gen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),	
<p>Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>	<ul style="list-style-type: none"> • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), 	<p><i>Beispiel 1: Kartenstapel beim Rommespiel</i></p> <p><i>Beispiel 2: Waffenarsenal im Heldenspiel</i></p>
<p>Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p> <p><i>LK: Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</i></p>	<ul style="list-style-type: none"> • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). <p><i>LK:</i></p> <ul style="list-style-type: none"> • <i>implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I),</i> 	<p><i>Beispiel: TODO-Liste</i></p> <p><i>Beispiel LK: Vokabeltrainer</i></p> <p>Eine beliebige Menge an Vokabeln soll erfasst werden können. Man sollte die Vokabeln nach einem bestimmten Wort durchsuchen können. Außerdem müssen die Operationen „Entfernen“ und „Hinzufügen“ möglich sein. Es können weitere Funktionalitäten, z.B. eine Sortierung nach Alphabet oder Schwierigkeit umgesetzt werden.</p>

3.3 Unterrichtsvorhaben III – Suchen und Sortieren

Thema: Suchen und Sortieren auf linearen Datenstrukturen (*LK: sowie Analyse und Beurteilung von Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen in kontextbezogenen Problemstellungen*)

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 (siehe schulinternes Curriculum Informatik Einführungsphase: Such und Sortieralgorithmen). Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

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

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 <ul style="list-style-type: none"> 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), 	<i>Beispiel 1:</i> Ländersuche als Memory <i>Beispiel 2:</i> Vokabelsuche in (un-)sortierte Liste
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	<ul style="list-style-type: none"> 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 	<i>Beispiel:</i> Karten sortieren

<p>(c) Entwicklung eines rekursiven Sortierverfahren für ein Feld (z.B. Sortieren durch Mischen)</p>	<p>men auch unter Verwendung von dynamischen Datenstrukturen (I),</p> <ul style="list-style-type: none"> • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), 	
<p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) Beurteilung der Effizienz der beiden Sortierverfahren</p> <p><i>LK untere Schranke für die Laufzeit von Sortieralgorithmen</i></p>	<ul style="list-style-type: none"> • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). <p><i>LK:</i></p> <ul style="list-style-type: none"> • <i>implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten)(I),</i> 	<p><i>Beispiel LK: Test- und Analyseumgebung für Sortieralgorithmen</i></p> <p>Listen mit unterschiedlich vielen Listenelementen und unterschiedlicher Vorsortierung werden bezüglich der Anzahl der Vergleiche von Listenelementen analysiert.</p> <p>Die untere Schranke für die Laufzeit wird bestimmt.</p>

3.4 Unterrichtsvorhaben IV – Relationale Datenbanken

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten (*LK: sowie Implementierung*)

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 (so weit nötig) normalisiert.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> Analyse vorgegebener SQL-Abfragen und 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> 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), 	<p><i>Beispiel 1: VideoCenter</i></p> <p>VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren. Unter http://dokumentation.videocenter.schule.de/old/video/index.html (abgerufen: 22. 10. 2018) findet man den Link zu</p>

<p>Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle</p> <ul style="list-style-type: none"> Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> bestimmen Primär- und Sekundärschlüssel (M), ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), überführen Datenbankschemata in vorgegebene Normalformen (M), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), 	<p>dem VideoCenter-System sowie nähere Informationen. Lesenswert ist auch die dort verlinkte „Dokumentation der Fallstudie“ mit didaktischem Material, welches alternativ bzw. ergänzend zu der im Folgenden beschriebenen Durchführung verwendet werden kann.</p> <p><i>Beispiel 2:</i> Schulbuchausleihe Unter www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php (abgerufen: 22. 10. 2018) 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.</p> <p><i>Beispiel 3:</i> Terradatenbank</p>
<p>2. Modellierung von relationalen Datenbanken</p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> 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 <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation 	<ul style="list-style-type: none"> 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). 	<p><i>Beispiel 1:</i> Buchungssystem 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. 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 22. 10. 2018) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt geht.</p> <p><i>Beispiel 2:</i> Schulverwaltung In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ableiten, dass z.B. Schüler X von Lehrer Y im 2. Halb-</p>

<ul style="list-style-type: none"> Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 		jahr des Schuljahrs 2015/2016 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.
--	--	---

3.5 Unterrichtsvorhaben V-GK – Sicherheit und Datenschutz in Netzstrukturen

Thema: Sicherheit und Datenschutz in Netzstrukturen

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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Kommunikation in Netzwerken</p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> 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 un- 	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken</p> <p><i>Beispiel:</i> Das Internetprotokollfamilie</p>
<p>2. Fallbeispiele zur Datenschutzproblematik</p>		<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Un-</p>

und zum Urheberrecht	ter Berücksichtigung unterschiedlicher Interessenlagen (A), <ul style="list-style-type: none"> • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D). 	<i>Lehrplandokument Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz Material der Fortbildung: Informatik, Mensch und Gesellschaft Beispiel: Der gläserne Mensch</i>
-----------------------------	---	--

3.6 Unterrichtsvorhaben V-LK – Softwareentwicklung mit Datenbankanbindung

LK-Thema: Projektorientierte Softwareentwicklung am Beispiel einer Anwendung mit Datenbankanbindung

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. Dann muss das *Unterrichtsvorhaben VI* zu nichtlinearen Datenstrukturen zuvor behandelt werden.

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 da-

bei 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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><i>Analyse einer lebensweltnahen Problemstellung im Hinblick auf die Entwicklung eines Java-Programms mit Datenbankbindung</i></p> <p><i>Entwicklung einer Programmidee</i></p> <p><i>Analyse des Problembereichs</i></p> <p><i>Entwicklung eines Anforderungskatalogs für das zu entwickelnde Programm</i></p>	<p><i>Die Schülerinnen und Schüler</i></p> <ul style="list-style-type: none"> • <i>ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</i> • <i>modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</i> • <i>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M),</i> • <i>ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</i> • <i>stellen die Kommunikation zwischen Objekten grafisch dar (D),</i> • <i>stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</i> • <i>dokumentieren Klassen (D),</i> • <i>analysieren und erläutern objektorientierte Modellierungen (A),</i> • <i>implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</i> • <i>ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</i> 	<p><i>Beispiel: Quizspiel</i> <i>Verwaltung von Quizfragen, Antworten und Ranglisten</i></p> <p><i>Beispiel: Navigationssystem</i> <i>Ermittlung von kürzesten Wegen im deutschen Autobahnnetz</i></p> <p><i>Beispiel: Verwaltung der Schülerbücherei</i> <i>Verwaltungsprogramm für das Einpflegen und Ausleihe von Büchern der Schülerbibliothek</i></p> <p><i>Beispiel: Fehlstundenverwaltung</i></p>

	<ul style="list-style-type: none"> • <i>stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</i> • <i>modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</i> • <i>bestimmen Primär- und Sekundärschlüssel (M),</i> • <i>implementieren ein relationales Datenbankschema als Datenbank (I),</i> • <i>analysieren und erläutern eine Datenbankmodellierung (A),</i> • <i>erläutern die Eigenschaften normalisierter Datenbankschemata (A),</i> • <i>überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),</i> • <i>überführen Datenbankschemata in die 1. bis 3. Normalform (M),</i> • <i>analysieren und erläutern Algorithmen und Programme (A),</i> • <i>modifizieren Algorithmen und Programme (I),</i> • <i>testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I),</i> • <i>ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</i> • <i>nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</i> • <i>beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</i> • <i>interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</i> • <i>analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</i> • <i>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),</i> • <i>wenden didaktisch orientierte Entwicklungsumgebungen zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</i> 	<p><i>Verwaltungsprogramm für die Fehlstunden von Schülerinnen und Schülern</i></p> <p><i>Beispiel: Sportfestverwaltung</i> <i>Verwaltung von Aufgaben, Sportereignissen und Ergebnissen des Schulsportfestes</i></p> <p><i>Beispiel: Materialverwaltung</i> <i>Materialien für Vertretungsstunden sollen verwaltet werden.</i></p>
--	---	---

- | | | |
|--|--|--|
| | <ul style="list-style-type: none"> • <i>entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M),</i> • <i>erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</i> • <i>untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A).</i> | |
|--|--|--|

3.7 Unterrichtsvorhaben VI – Dynamische, nichtlineare Datenstrukturen

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

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

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

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

LK: 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 Algo-

rithmen 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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p> <p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p> <p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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), • 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), 	<p><i>Material:</i> Leitprogramm von EducETH Zürich</p> <p><i>Beispiel 1:</i> Spielbäume (Streichholzspiel und TicTacTo)</p> <p><i>Beispiel 2:</i> Entscheidungsbäume 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.</p> <p><i>Beispiel 3:</i> Codierungsbäume für Morsecode <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum</p> <p><i>Beispiel 4:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Beispiel 5:</i> Informatikerbaum als Suchbaum In einem binären Suchbaum werden die Namen</p>

<p>Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>	<ul style="list-style-type: none"> • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). 	<p>und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert.</p>
<p>LK:</p> <p>1. Analyse von Graphen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Graph, gerichtet - ungerichtet, Knoten, Kanten, Kantengewicht)</p> <p>(b) Aufbau und Darstellung von Graphen anhand von Graphenstrukturen in verschiedenen Kontexten (Adjazenzmatrix, Adjazenzliste)</p> <p>2. Die Datenstruktur Graph im Anwendungskontext unter Nutzung der Klassen Graph, Vertex und Edge.</p> <p>(a) Erarbeitung der Klassen Graph, Vertex und Edge und beispielhafte Anwendung der Operationen</p> <p>(b) Bestimmung von Wegen in Graphen im Anwendungskontext (Tiefensuche, Breitensuche)</p>	<p>LK:</p> <ul style="list-style-type: none"> • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Backtracking“(M), • entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M), • implementieren Operationen dynamischer Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • wenden didaktisch orientierte Entwicklungsumgebungen zur Demonstration, zum Ent- 	<p><i>Beispiel 1: Soziale Netzwerke</i></p> <p><i>Beispiel 2: Wegsuche</i> Tiefensuche mit Backtracking. 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 Vorgängersuche in dieser Liste ein Weg vom Start- zum Zielknoten gefunden werden. Damit wird das Verfahren beim Dijkstra-Algorithmus vorbereitet.</p> <p><i>Beispiel: Kürzeste Wege</i> Ausgehend vom Backtracking-Algorithmus zur Bestimmung aller Wege von einem Start- zu einem Zielknoten in einem Graphen wird ein Algo-</p>

<p>che)</p> <p>(c) <i>Bestimmung von kürzesten Wegen in Graphen im Anwendungskontext (Backtracking, Dijkstra).</i></p> <p><i>Bestimmung von minimalen Spannbäumen eines Graphen im Anwendungskontext.</i></p>	<p>wurf, zur Implementierung und zum Test von Informatiksystemen an (I),</p> <ul style="list-style-type: none"> • <i>nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),</i> • <i>nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K).</i> 	<p>rithmus zur Bestimmung des kürzesten Weges erarbeitet.</p> <p><i>Aufwandbetrachtungen führen zu der Frage nach einem effizienteren Algorithmus -> Dijkstra-Algorithmus. Die Implementierung erfolgt in der Lerngruppe arbeitsteilig unter Vorgabe einer Benutzungsoberfläche. Der Vergleich der beiden Algorithmen unter Effizienzaspekten ist Bestandteil des Unterrichts.</i></p>
---	--	---

3.8 Unterrichtsvorhaben VII – Automaten und formale Sprachen

Thema: Endliche Automaten und formale Sprachen

Anhand kontextbezogener Beispiele werden endliche Automaten innerhalb eines Stationenlernens 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.

LK: Blick auf diese Einschränkungen endlicher Automaten wird die Idee eines Automaten mit Speicher thematisiert und zu einem Kellerautomaten weiterentwickelt und kontextfreie Grammatiken behandelt. Als Anwendung wird ein Parser modelliert und implementiert.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p> <p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p> <p>3. Grenzen endlicher Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), analysieren und erläutern Grammatiken regulärer Sprachen (A), zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), modifizieren Grammatiken regulärer Sprachen (M), entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), 	<p><i>Materialien:</i> Stationenlernen</p> <p><i>Beispiel:</i> Brandgefahr Abituraufgabe 2012</p> <p><i>Beispiel:</i> Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$</p>
<p>LK:</p> <p>Entwicklung eines Kellerautomaten als Antwort auf die Grenzen endlicher Automaten</p> <p>(a) Erweiterung eines DEA um eine einzelne Speichervariable zum Zählen von Eingabezeichen (z.B. Klammern) und Problematisierung</p>		

<p><i>dieses Ansatzes</i></p> <p>(b) <i>Entwicklung eines Automaten mit Kellerspeicher</i></p> <p>(c) <i>Anwendung eines Kellerautomaten zur Syntaxüberprüfung auf Grundlage von nicht-regulären Grammatiken</i></p> <p>(d) <i>Implementierung eines Kellerautomaten zur Syntaxüberprüfung (Backtracking)</i></p>	<ul style="list-style-type: none"> • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D). <p>LK: <i>die Kompetenzen beziehen sich zusätzlich auf Kellerautomaten und kontextfreie Grammatiken.</i></p>	
---	---	--

3.9 Unterrichtsvorhaben VIII-GK – Grenzen der Automatisierbarkeit

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

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, das 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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <ul style="list-style-type: none"> 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 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung</p> <p><i>Materialien 2:</i> Rollenspiel Von-Neumann-Architektur</p>
<p>2. Grenzen der Automatisierbarkeit</p> <ul style="list-style-type: none"> a) Vorstellung des Halteproblems b) Unlösbarkeit des Halteproblems c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen 		<p><i>Beispiel:</i> Halteproblem</p>

3.10 Unterrichtsvorhaben VIII-LK – Arbeitsweise eines Computers

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

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 VII – Automaten und formale Sprachen* in Beziehung gesetzt.

Ausgehend von einer einfachen formalen Sprache werden die Bestandteile eines Compilers dargestellt: Der Scanner eines Compilers wird in Form eines endlichen Automaten modelliert und implementiert. 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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>(a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>(b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>(c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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 Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), 	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 -Von-Neumann-Architektur und maschinennahe Programmierung</p> <p><i>Materialien 2:</i> Rollenspiel Von-Neumann-Architektur</p>

<p>2. Simulation der Phasen eines Compilers</p> <p>(a) Prozesse beim Compiler:</p> <ul style="list-style-type: none"> • scannen • parsen • übersetzen/interpretieren <p>(b) Arten von Fehlern:</p> <ul style="list-style-type: none"> • lexikalischer Fehler • syntaktischer Fehler • semantischer Fehler <p>(c) Einordnung der neuen Begriffe in den Gesamtkontext der formalen Sprachen</p> <ul style="list-style-type: none"> • Automaten • Grammatiken • Sprachen <p>3. Die Schritte eines Compilers</p> <p>(a) Scanner:</p> <ul style="list-style-type: none"> • 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 <p>(b) Parser:</p> <ul style="list-style-type: none"> • reguläre (oder wahlweise kontextfreie) Grammatik als Grundlage 	<ul style="list-style-type: none"> • 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), • 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), • modifizieren Grammatiken regulärer und kontextfreier Spra- 	<p><i>Material LK: Vom Text zum Programm, Scanner und Parser „Informatik 2“ (S.160ff)</i></p>
---	---	---

<ul style="list-style-type: none"> • Vorgabe einer Grundversion des Parsers • Erweiterung des Sprachumfangs • Implementierung der Parsermethoden für die Produktionsregeln der kontextfreien Grammatik <p>(c) Interpreter</p> <ul style="list-style-type: none"> • Vorgabe einer Grundversion des Interpreters • Erweiterung des Sprachumfangs • Implementierung 	<p>chen (M),</p> <ul style="list-style-type: none"> • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M), • 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). 	
--	---	--

3.11 Unterrichtsvorhaben IX-LK – Grenzen und Auswirkung der Automatisierbarkeit

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

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, in wieweit 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.

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 Daten zu schützen, wird im zweiten Teil die Umsetzung 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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Einführung in die Problemfelder Datenschutz, Urheberrecht und moralische Verantwortung</p> <p>a. Vorstellung eines komplexen Fallbeispiels</p> <p>b. Erarbeitung von Interesse verschiedener Interessensgruppen im Hinblick auf die Problemfelder</p> <p>c. Erster Bewertungsversuch auf Grundlage des Vorwissens von Schülerinnen und Schülern</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A). • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A). 	<p><i>Beispiel: autonomes Fahren</i></p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I - Fallbeispiel „autonomes Fahren“</p>
<p>2. Erarbeitung grundlegender Positionen (ggf. in zieldifferenten Gruppen)</p> <p>a. Grundideen des Datenschutzes</p> <p>b. Urheberrechts</p> <p>c. Erarbeitung eines einfachen moralischen Bewertungsmaßstabes</p> <p>d. Anwendung auf Fallbeispiele</p>	<ul style="list-style-type: none"> • analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A). 	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I - Grundlagen „Informatik, Mensch und Gesellschaft“</p> <p><i>Beispiel: Creative-Commons-Lizenzen</i></p>
<p>3. Zusammenfassung und Sicherung</p> <p>a. Darstellung der relevanten Aspekte zu 2.</p> <p>b. Demonstration an jeweils einem einfachen Beispiel</p> <p><i>Anmerkung:</i> Dieser Schritt ist entscheidend, wenn</p>	<ul style="list-style-type: none"> • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D). 	<p><i>Beispiel: Stellwände</i></p> <p>Eine Zusammenfassung und Sicherung kann in Form von Plakatstellwänden erfolgen. Diese können auch im Sinne einer Dauerausstellung im Schulgebäude präsentiert werden.</p>

<p>vorher arbeitsteilig gearbeitet wurde.</p>		
<p>4. Stellungnahme zu einem komplexen Fallbeispiel</p> <ul style="list-style-type: none"> a. Reflexion des Einstiegsbeispiels des Unterrichtsvorhabens auf Grundlage der erarbeiteten Positionen b. Einführung in das Verfassen von Stellungnahmen c. Erarbeitung von Stellungnahmen zum Einstiegsbeispiel d. Diskussion mit Blick auf Grenzen der Automatisierbarkeit 		<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I - Materialblatt „Stellungnahme“</p>
<p>5. Sicherheitsziele und der kryptographische Ansatz</p> <ul style="list-style-type: none"> 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 		<p><i>Beispiel:</i> Sicherheitsziele <i>Zugriffskontrolle</i> und <i>Authentifikation</i> anhand des „Keyless Entry-System“ bei Autos</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.I - Sicherheitsziele und Kommunikationsrollenspiel</p>

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

Beispiele: Caesar-/Vigenère-Verschlüsselung/
One-Time-Pad / RSA-Verfahren

Materialien: <https://www.cryptool.org/de/>
(Abgerufen am 22.10.2018)

3.12 Unterrichtsvorhaben X-LK – Netzwerkkommunikation

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

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 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.

Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Grundlagen der Datenübertragung in Netzwerken</p> <p>(a) Schichtenmodell (b) Grundlagen der Codierung (c) Topologien (d) Routing Analyse von Grundlagen der Adressierung in IP-Netzwerken (e) Protokolle</p> <p>2. Analyse, Modellierung und Implementierung von Netzwerkanwendungen in Client-Server-Struktur</p> <p>(a) Nutzung einfacher Server-Dienste mittels Client (b) Anbieten von Diensten mittels Server (c) Modellierung und Implementierung eigener Server</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern 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), • modellieren Klassen mit ihren Attributen, Methoden und Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), 	<p><i>Material 1: Filius</i></p> <p><i>Material: Aufgabensammlung New Town und Gold City</i></p>
<p>3. Entwicklung eines vollständigen Client-Server-Systems</p> <ul style="list-style-type: none"> • Protokollentwurf, Dialogorientierung • Modellierung mittels Entwurfs- und Implementationsdiagramm • Bedeutung von Nebenläufigkeit • Implementierung 	<ul style="list-style-type: none"> • 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). 	<p><i>Beispiel: Messenger-Dienst</i></p> <p>Entwickeln eines Client-Server-Systems zum Versenden von Nachrichten zwischen einzelnen Rechnern (einfacher Messenger), basierend auf selbst gewählten „Nicknames“.</p>