

---

Kreisgymnasium Halle

# **Schulcurriculum Informatik**

**Qualifikationsphase – Grundkurs**

---

Stand: 09.07.2016

## Konkretisierte Unterrichtsvorhaben

Dieses Schulcurriculum hat den Anspruch, alle im Kernlehrplan geforderten Kompetenzen abzubilden.

Die Umsetzung des Lehrplans folgt pro Unterrichtsvorhaben jeweils auf einer Übersichts- und einer dazugehörigen Konkretisierungsebene. Die Übersichtsebene ist lehrwerksunabhängig formuliert, die dargestellten Konkretisierungen beziehen sich auf das aktuell eingesetzte Schulbuch *Informatik 2* des Verlags Schöningh.

Der je Unterrichtsvorhaben angegebene Stundenumfang versteht sich als grobe Orientierung, von der in der Unterrichtspraxis abgewichen werden kann.

### Didaktische Lernumgebung

Die Unterrichtsvorhaben in Informatik hängen in ihrer konkreten Umsetzung unter anderem von den gewählten Programmierumgebungen ab.

Es sei an dieser Stelle ausdrücklich darauf hingewiesen, dass die Lehrkraft selbstverständlich die Freiheit hat, *abweichende*, z.B. *neuere* als die im Lehrbuch oder Schulcurriculum erwähnten Softwareumgebungen zur Erreichung der Lernziele und Förderung der jeweils relevanten Kompetenzen einzusetzen.

## Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase - Q1

Qualifikationsphase – Q1	
<p><b><u>Unterrichtsvorhaben Q1-I</u></b></p> <p><b>Thema:</b> Wiederholung und Vertiefung der objektorientierten Modellierung</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Darstellen und Interpretieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Informatik, Mensch und Gesellschaft</li> <li>- Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Wirkung der Automatisierung</li> <li>- Nutzung von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> 14 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q1-II</u></b></p> <p><b>Thema:</b> Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Wirkungen der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>
<p><b><u>Unterrichtsvorhaben Q1-III</u></b></p> <p><b>Thema:</b> Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Wirkungen der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 16 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q1-IV</u></b></p> <p><b>Thema:</b> Automaten und formale Sprachen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Formale Sprachen und Automaten</li> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Endliche Automaten</li> <li>- Grammatiken regulärer Sprachen</li> <li>- Möglichkeiten und Grenzen von Automaten und formalen Sprachen</li> <li>- Einzelrechner und Rechnernetzwerke</li> <li>- Grenzen der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 24 Stunden</p>
<p><b>Summe Qualifikationsphase 1: 74 Stunden</b></p>	

## Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase – Q2

Qualifikationsphase – Q2	
<p><b><u>Unterrichtsvorhaben Q2-I</u></b></p> <p><b>Thema:</b> Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q2-II</u></b></p> <p><b>Thema:</b> Aufbau von und Kommunikation in Netzwerken</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Einzelrechner und Rechnernetzwerke</li> <li>- Sicherheit</li> <li>- Nutzung von Informatiksystemen, Wirkungen der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 16 Stunden</p>
<p><b><u>Unterrichtsvorhaben Q2-III</u></b></p> <p><b>Thema:</b> Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Datenbanken</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Sicherheit</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>	
<p><b>Summe Qualifikationsphase 2: 56 Stunden</b></p>	

Die folgende Kompetenz wird in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und soll aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler ...

- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K).

### **Unterrichtsvorhaben Q1-I**

**Thema:** Wiederholung und Vertiefung der objektorientierten Modellierung

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

#### **Vorhabenbezogene Konkretisierung:**

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

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

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

Das Paradigma der objektorientierten Modellierung wird um das Konzept der abstrakten Klasse erweitert.

**Zeitbedarf:** 14 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</b></p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>- analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> </ul>	<p><b>Kapitel 1 Konzepte des objektorientierten Modellierens</b></p> <p>1.1 Modellierung der Realität</p> <p>1.2 Die Welt ist voller Objekte Projekteinstieg: Klassentwurf – step by step</p>
<p><b>2. Erweiterung der objektorientierten Programmierung</b></p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> </ul>	<p>1.3 Gut geplant – Klassentwurf</p> <p>1.4 Hierarchien machen's einfacher – Vererbung</p>
<p><b>3. Mensch und Technik</b></p> <p>a) Informatiker verändern die Welt</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> <li>- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen (D),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</li> <li>- 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).</li> </ul>	<p>Die digitale Welt 001 – Mensch und Technik</p>

## Unterrichtsvorhaben Q1-II

### **Thema:**

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

### **Leitfragen:**

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

### **Vorhabensbezogene Konkretisierung:**

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

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

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

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

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

<p><b>4. Die Datenstruktur Liste</b></p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		<p>2.5 Flexibel für alle Fälle – lineare Listen</p>
<p><b>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</b></p>		<p>2.6 Prüfungsvorbereitung</p>
<p><b>6. Datenschutz</b></p> <p>a) Datenschutz als Grundrecht</p> <p>b) Das Datenschutzgesetz</p> <p>c) Datensammler</p>		<p><i>Projekteinstieg: Wartende Helden</i></p> <p><i>Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.</i></p> <p>Die digitale Welt 101 – Datenschutz</p>

## Unterrichtsvorhaben Q1-III

**Thema:** Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es werden, im Rückgriff auf die Vorkenntnisse aus der Einführungsphase, mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie deren Effizienz beurteilt.

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Eigenschaften von Algorithmen</b></p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> </ul>	<p><b>Kapitel 3 Algorithmen</b></p> <p>3.1 Ohne Algorithmen läuft nichts</p> <p>3.2 Teile die Arbeit – rekursive Algorithmen</p>
<p><b>2. Suchen in Listen und Arrays</b></p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> <li>- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> </ul>	<p>3.3 Suchen – iterativ und rekursiv</p> <p>Projekteinstieg: Laufzeitanalyse experimentell</p>
<p><b>3. Sortieren auf Listen und Arrays</b></p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen)</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p> <p>d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</li> <li>- 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).</li> </ul>	<p>3.4 Sortieren – iterativ und rekursiv</p>
<p><b>4. Verantwortung der Informatik</b></p> <p>a) Der Unterschied zwischen Anwender und Produzent von Informatiksystemen</p> <p>b) Informatik und Ethik</p>		<p>Die digitale Welt 011 – Verantwortung der Informatik</p>

## Unterrichtsvorhaben Q-I IV

**Thema:** Automaten und formale Sprachen

### **Leitfragen:**

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen beschrieben werden und welche nicht? Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?

### **Vorhabensbezogene Konkretisierung:**

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

An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht.

**Zeitbedarf:** 20 Std.

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Endliche Automaten</b></p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEAs durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A),</li> <li>- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D),</li> <li>- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>- stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> <li>- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>- analysieren und erläutern Grammatiken regulärer Sprachen (A),</li> <li>- modifizieren Grammatiken regulärer Sprachen (M),</li> <li>- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),</li> </ul>	<p><b>Kapitel 4 Endliche Automaten und formale Sprachen</b></p> <p>4.1 Endliche Automaten Projekteinstieg: Schatzsuche</p>
<p><b>2. Grammatiken regulärer Sprachen</b></p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> <li>- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M)</li> <li>- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M),</li> <li>- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</li> <li>- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),</li> </ul>	<p>4.2 Formale Sprachen</p>
<p><b>3. Übungen und Vertiefungen</b></p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>	<ul style="list-style-type: none"> <li>- erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),</li> <li>- untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<p>4.5 Prüfungsvorbereitung</p>
<p><b>4. Grundsätzliche Arbeitsweise eines Computers und Grenzen der Berechenbarkeit</b></p> <p>a) Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>b) Grenzen der Berechenbarkeit anhand des Halteproblems, nicht effizient berechenbare Probleme</p>		<p>Die digitale Welt 101 – Maschinennahe Programmierung</p> <p>Die digitale Welt 100 – Berechenbarkeit</p>

## Unterrichtsvorhaben Q2-I

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

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

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

**Zeitbedarf:** 20 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Aufbau von Baumstrukturen und Grundbegriffe</b></p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<p><b>Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p><b>2. Binäre Bäume</b></p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>- verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> </ul>	<p>5.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p><b>3. Binäre Suchbäume</b></p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I).</li> </ul>	<p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>		<p>5.8 Prüfungsvorbereitung</p>

## Unterrichtsvorhaben Q2-II

**Thema:** Aufbau von und Kommunikation in Netzwerken

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

**Vorhabenbezogene Konkretisierung:**

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

**Zeitbedarf:** 12 Stunden

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</b> a) Kommunikation im Shannon-Weaver-Modell b) Kriterien von technischen Kommunikationsarten c) Die Geschichte der technischen Kommunikation	Die Schülerinnen und Schüler ... - beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	<b>Kapitel 6 Kommunikation in Netzwerken</b> 6.1 Menschen kommunizieren – ohne und mit Technik Projekteinstieg: Kommunikation im Wilden Westen
<b>2. Aufbau von Netzwerken und Kommunikationsregeln</b> a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen		6.2 Ohne Protokoll läuft nichts – Netzwerke
<b>3. Aufgabenteilung in Netzwerken durch Server und Client</b> a) Aufbau und Aufgaben der Client-Server-Struktur b) Protokolle zwischen Client und Server		6.3 Einer für alle – Client-Server-Struktur
<b>4. Kryptologie</b> a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA) b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand		Die digitale Welt 111 – Kryptologie
<b>5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</b>		6.4 Prüfungsvorbereitung

## Unterrichtsvorhaben Q2-III

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

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

**Zeitbedarf:** 20 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Nutzung von relationalen Datenbanken</b> a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen b) SQL-Abfragen - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN)	Die Schülerinnen und Schüler ... - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),	<b>Kapitel 8 Datenbanken</b> 7.1 Wissen speichern und verwalten – Datenbanksysteme 7.2 Daten anordnen mit Tabellen Projekteinstieg: Tabellen 7.3 Daten filtern mit SQL 7.4 Komplexe Filter

<ul style="list-style-type: none"> <li>- Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke)</li> </ul> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<ul style="list-style-type: none"> <li>- modifizieren eine Datenbankmodellierung (M),</li> <li>- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>- bestimmen Primär- und Sekundärschlüssel (M),</li> <li>- analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>- erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li> <li>- überführen Datenbankschemata in die 1. bis 3. Normalform (M).</li> </ul>	
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> <li>- Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>- Erläuterung und Erweiterung einer Datenbankmodellierung</li> </ul> <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>- Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln</li> </ul> <p>c) Normalformen</p> <ul style="list-style-type: none"> <li>- Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>		<p>7.5 Datenbankentwurf</p> <p>7.6 Umsetzung des ER-Modells</p> <p>Wiederaufgriff des Projekteinstiegs</p> <p>7.7 Datenbanken verbessern durch Normalformen</p>
<p><b>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b></p>		<p>7.8 Prüfungsvorbereitung</p>