

II. Qualifikationsphase Q1

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

Die Schülerinnen und Schüler

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

Im ersten Halbjahr 2 Klausuren, im 2. Halbjahr eine Klausur oder eine Facharbeit und ein Projekt. Die Länge der Klausuren beträgt 90 min.

Unterrichtsvorhaben Q1- I:

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

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

Vorhabenbezogene Konkretisierung:

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

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

Zeitbedarf: 8 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. 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</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern objektorientierte Modellierungen, • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten, • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu, • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren, • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken, • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informationssystemen an, • interpretieren Fehlermeldungen und korrigieren den Quellcode, • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar, • dokumentieren Klassen, • stellen die Kommunikation zwischen Objekten grafisch dar. 	<p><i>Beispiel:</i> Die Geburt der Frösche → Java-Klassendefinition mit Greenfoot (AB 11)</p> <p><i>Beispiel:</i> Gras und Schafe (und Wölfe und Wetter und ...) → Arbeitsblätter 1 - 3 vorhanden (ausbaufähig für ein Projekt als Alternative zur 2. Klausur, Arbeitsblatt vorhanden)</p> <p><i>Beispiel:</i> Wetthüpfen Für ein Wetthüpfen zwischen einem Hasen, einem Hund und einem Vogel werden die Tiere gezeichnet. Alle Tiere springen wiederholt nach links. Die Höhe und Weite jedes Hüpfers ist zufällig. Evtl. marschieren sie anschließend hintereinander her. oder</p> <p><i>Beispiel:</i> Tannenbaum Ein Tannenbaum soll mit verschiedenen Arten von Schmuckstücken versehen werden, die durch unterschiedliche geometrische Objekte dargestellt werden. Es gibt Kugeln, Päckchen in der Form von Würfeln und Zuckerringe in Form von Toren. Ein Prototyp, der bereits mit Kugeln geschmückt werden kann, kann zur Verfügung gestellt werden. Da alle Schmuckstücke über die Funktion des Auf- und Abschmückens verfügen sollen, liegt es nahe, dass entsprechende Methoden in einer gemeinsamen Oberklasse realisiert werden. <i>Materialien:</i> <i>Greenfoot</i></p>

Unterrichtsvorhaben Q1- II:

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

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen, • analysieren und erläutern Algorithmen und Programme, • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu, • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen, • modifizieren Algorithmen und Programme, • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen, • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, • interpretieren Fehlermeldungen und korrigieren den Quellcode, • testen Programme systematisch anhand von Beispielen, • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau 	<p>Klett Informatik 4 S.10 – 15</p> <p><i>Beispiel: Warteschlangen</i></p> <p>konkret: Taxiwarteschlange in Greenfoot mit Feldern</p> <p>Anwendung: Schülerschlange am Kiosk mit Feldern</p>

<p>2. Die Datenstruktur 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>(c) Rekursive Funktionen</p> <p>(d) Rekursive Methoden einer Liste</p>	<ul style="list-style-type: none"> • vergleichen die Datenstrukturen Warteschlange und Stapel • sind mit dem LIFO- und FIFO-Prinzip vertraut 	<p>Klett Informatik 4 S.16 – 31</p> <p><i>Beispiel: Warteschlangen</i></p> <p>konkret: Taxiwarteschlange in Greenfoot mit Listen</p> <p>Anwendung: Schülerschlange am Kiosk mit Listen</p> <p><i>Beispiel:</i></p> <ul style="list-style-type: none"> ➔ Kaninchenproblem (Fibonacci) ➔ Reiskornparabel (Film)
<p>3. 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>		<p>Klett Informatik 4 S.32 – 36</p> <p><i>Beispiel:</i></p> <p>UNO – ein beliebtes Kartenspiel</p> <p>Der Eisbär</p> <p>Schwerpunkt:</p> <ul style="list-style-type: none"> ➔ Türme von Hanoi ➔ (www.blinde-kuh.de/spiele/hanoi)
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p> <p>(a) Heterogene Listen</p> <p>(b) Sortierte Listen</p>		<p><i>Beispiel:</i></p> <p>Der Biber im Restaurant (aus Schöningh „Informatik 2“ S.73)</p>

Unterrichtsvorhaben Q1- III:

Thema: „Ich habe ein Problem ..., denn Ordnung muss sein!“

Leitfrage: Wie kann man gespeicherte Informationen günstig (wieder-)finden? Suchen und Sortieren auf linearen Datenstrukturen.

Vorhabenbezogene Konkretisierung:

Zuerst wird die Problematik von Bedienungsanleitungen, Sprache als Beschreibung und Überführung in konkrete Algorithmen behandelt und vertieft. Hierbei wird der Einsatz von Struktogrammen und Programmablaufplänen wiederholt, Kontrollstrukturen vertieft.

In einem Anwendungskontext werden 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.

In diesem Lernabschnitt steht das eigenverantwortliche Arbeiten (kurz EVA) im Vordergrund. Durch Ausprobieren sollen Sortierverfahren erkannt und umgesetzt werden. Die neuen Erkenntnisse werden algorithmisch formuliert und vom Pseudocode über das Struktogramm in einen Javacode überführt.

Zeitbedarf: 16 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Algorithmen – Methoden und Struktur</p> <p>(a) Bedienungsanleitungen (b) Merkmale von Algorithmen (c) Pseudocode – Struktogramm/PAP – Implementierung (d) Vertiefung der Kontrollstrukturen (e) Wie löst man komplexe Probleme?</p> <p>2. Suchen von Daten in Listen und Arrays</p> <p>(f) Lineare Suche in Listen und in Arrays (g) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (h) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern Algorithmen und Programme, beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen, beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen, entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“, modifizieren Algorithmen und Programme, implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen, 	<p>Klett Informatik 3 S.17 – 24</p> <p>Kontrollstrukturen aus: <i>Schöningh „Informatik 1“ S.59 – 69 neue Ausgabe</i> → „Der Rover hat ein Problem“</p> <p><i>Schöningh „Informatik 2“ S.94 – 95</i> → Projekteinstieg 1 „Suchen“</p>
<p>3. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</p>	<ul style="list-style-type: none"> implementieren und erläutern iterative und rekursive Such- und Sortierverfahren, nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen, interpretieren Fehlermeldungen und korrigieren den Quellcode, testen Programme systematisch anhand von Beispielen, stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar. 	<p><i>Schöningh „Informatik 2“ S.96 – 97</i> → Projekteinstieg 2 „Sortieren“</p>

Unterrichtsvorhaben Q1- IV:

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: 20 Stunden

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 Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<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, analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage, analysieren und erläutern eine Datenbankmodellierung, erläutern die Eigenschaften normalisierter Datenbankschemata, bestimmen Primär- und Sekundärschlüssel, ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten, modifizieren eine Datenbankmodellierung, modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema, bestimmen Primär- und Sekundärschlüssel, überführen Datenbankschemata in vorgegebene Normalformen, verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren, ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen, stellen Entitäten mit ihren Attributen und die 	<p>EVA – Eigenverantwortliches Arbeiten</p> <p>Vieweg/Teubner „Informatik – Manga Datenbanken“</p>

2. Modellierung von relationalen Datenbanken

(a) Entity-Relationship-Diagramm

- Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms
- Erläuterung und Modifizierung einer Datenbankmodellierung

(b) Entwicklung einer Datenbank aus einem Datenbankentwurf

- Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln

(c) Redundanz, Konsistenz und Normalformen

- Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation
- Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)

Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar,

- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften.

Unterrichtsvorhaben Q1- V:

Thema: Sicherheit und Datenschutz in Netzstrukturen

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

Vorhabenbezogene Konkretisierung:

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

Zeitbedarf: 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</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, • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren, • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts, • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen, • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte. 	<p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1)</p> <p>Verschiedene Fernsehdokumentationen</p> <ul style="list-style-type: none"> ➔ „Gestohlene Identität“ (MDR) ➔ „Krieg der Drohnen“ (ARTE) ➔ „NetWars – Krieg im Netz“ (ARTE)“
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>		<p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz (Download Q1-V.2)</p>

