



# Informatik

---

Stand: 21.08.2016

## 1. Rahmenbedingungen der fachlichen Arbeit

Das Fach Informatik wird am FvS-Gymnasium in der Sekundarstufe II als dreistündiger Grundkurs angeboten, allgegenwärtig in Wirtschaft, Gesellschaft, Arbeit und Freizeit hat es einen großen Anteil am Entwicklungsstand unserer technisierten und globalisierten Welt.

Der Informatikunterricht der gymnasialen Oberstufe geht über eine Grundbildung im Bereich der Informations- und Kommunikationstechnologien in der Sekundarstufe I hinaus. Die Schülerinnen und Schüler erwerben über rezeptive Medienanwendungen und die interaktive Nutzung von Medienangeboten in den Fächern hinaus Fähigkeiten zur kritischen und verantwortungsvollen Analyse, Modellierung und Implementierung komplexer Informatiksysteme. Dabei konzentriert sich der Unterricht stets auf fundamentale und zeitbeständige informatische Ideen, Konzepte und Methoden und schließt auch die Auseinandersetzung mit Fragen einer menschengerechten Gestaltung und der Sicherheit von Systemen sowie der Folgen und Wirkungen des Einsatzes von Informatiksystemen ein. Schülerinnen und Schüler werden so befähigt und motiviert, auch zukünftige Entwicklungen zu nutzen, zu verstehen, hinsichtlich ihrer Wirkungen zu beurteilen und sich aktiv an der Fortentwicklung zu beteiligen. (Kernlehrplan Informatik für die Sekundarstufe II, Gymnasium / Gesamtschule in Nordrhein-Westfalen S. 10-11)

Ausgangspunkt im Informatikunterricht ist häufig ein Problem mit lebensweltlichem Bezug. Schülerinnen und Schüler erwerben und erweitern in der aktiven Auseinandersetzung mit komplexen Problemstellungen Kompetenzen, die sie zum selbstständigen informatischen Problemlösen befähigen. Mit der Aneignung von Strategien und Techniken zur strukturierten Zerlegung im Problemlöseprozess, zur Algorithmisierung von Abläufen sowie zur formalsprachlichen, grafischen oder symbolischen Beschreibung von Sachverhalten und Zusammenhängen erwerben die Lernenden Kompetenzen zur Bewältigung von Komplexität. Die Konstruktion eines abstrakten Modells zu einer anwendungsbezogenen Problemstellung fördert das Abstraktionsvermögen sowie kreatives und strukturelles Denken. Die Umsetzung eines informatischen Modells in ein lauffähiges Informatiksystem hat für Schülerinnen und Schüler nicht nur einen hohen Motivationswert, sondern ermöglicht ihnen auch die eigenständige Überprüfung der Angemessenheit und Wirkung des Modells im Rückbezug auf die Problemstellung. Im Unterricht lassen sich umfangreiche Informatiksysteme nur in arbeitsteiliger projektorientierter Zusammenarbeit im Team erstellen. Solche Projekte können nur gelingen, wenn die gemeinsame Arbeit strukturiert geplant und organisiert wird. Insgesamt leistet das Fach Informatik in der gymnasialen Oberstufe damit einen wichtigen Beitrag zu einer erweiterten Allgemeinbildung und allgemeinen Studierfähigkeit der Schülerinnen und Schüler.

Die inhaltliche und methodische Gestaltung des Unterrichts ist entscheidend dafür, dass Schülerinnen und Schüler die ausgewiesenen Kompetenzen erwerben können. Informatikunterricht erfordert in hohem Maße die Anwendung schüleraktivierender Methoden, die selbstständiges Lernen ermöglichen und individuelle Förderung begünstigen. Unterschiedliche, auch geschlechtsspezifische Herangehensweisen, Interessen, Vorerfahrungen und fachspezifische Kenntnisse sind angemessen zu berücksichtigen.

In der Einführungsphase werden ausgehend von einfachen Fragestellungen und unter Anleitung der Lehrperson zunächst einzelne Stufen eines Problemlösungsprozesses durchlaufen. (Kernlehrplan Informatik für die Sekundarstufe II, Gymnasium / Gesamtschule in Nordrhein-Westfalen S. 11)



## Einführungsphase EF

### 2. Übersichtsraster Unterrichtsvorhaben in der Einführungsphase Teil 1

Einführungsphase	
<p><b><u>Unterrichtsvorhaben E-I</u></b></p> <p><b>Thema:</b> <i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</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>- Einsatz, Nutzung und Aufbau von Informatiksystemen</li> <li>- Wirkung der Automatisierung</li> </ul>	<p><b><u>Unterrichtsvorhaben E-II</u></b></p> <p><b>Thema:</b> <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></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>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</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>- Syntax und Semantik einer Programmiersprache</li> </ul>
<p><b><u>Unterrichtsvorhaben E-III</u></b></p> <p><b>Thema:</b> <i>Algorithmische Grundstrukturen in Java</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</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> </ul>	<p><b><u>Unterrichtsvorhaben E-IV</u></b></p> <p><b>Thema:</b> <i>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</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>- Binäre Codierung und Verarbeitung</li> </ul>



<ul style="list-style-type: none"><li>- Syntax und Semantik einer Programmiersprache</li><li>- Analyse, Entwurf und Implementierung einfacher Algorithmen</li></ul>	<ul style="list-style-type: none"><li>- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten</li></ul>
---	--



Übersichtsraster Unterrichtsvorhaben in der Einführungsphase Teil 2

<b>Einführungsphase</b>	
<p><b><u>Unterrichtsvorhaben E-V</u></b></p> <p><b>Thema:</b> <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>- Kommunizieren und Kooperieren</li><li>- Darstellen und Interpretieren</li><li>- Argumentieren</li><li>- Modellieren</li><li>- Implementieren</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>- Syntax und Semantik einer Programmiersprache</li><li>- Analyse, Entwurf und Implementierung einfacher Algorithmen</li></ul>	<p><b><u>Unterrichtsvorhaben E-VI</u></b></p> <p><b>Thema:</b> <i>Such- und Sortieralgorithmen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>- Argumentieren</li><li>- Modellieren</li><li>- Darstellen und Interpretieren</li><li>- Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>- Algorithmen</li><li>- Daten und ihre Strukturierung</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>- Algorithmen zum Suchen und Sortieren</li><li>- Analyse, Entwurf und Implementierung einfacher Algorithmen</li><li>- Objekte und Klassen</li></ul>
<p><b><u>Unterrichtsvorhaben E-VII</u></b></p> <p><b>Thema:</b> <i>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>- Kommunizieren und Kooperieren</li><li>- Darstellen und Interpretieren</li><li>- Argumentieren</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>- Geschichte der automatischen Datenverarbeitung</li><li>- Wirkungen der Automatisierung</li></ul>	



- Dateisystem	
---------------	--

Die Angaben zu den Kapiteln und Materialien beziehen sich in allen Sequenzen in Schwarz auf die neuere Ausgabe der Reihe „Informatik“ aus dem Schöningh-Verlag, in Blau auf die ältere Ausgabe erschienen 2010-2013. Steht vor der Kapitelnummer eine weitere Zahl, so bezieht sich diese auf den Band.

### **Unterrichtsvorhaben EF-I**

**Thema:** Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik

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

#### **Vorhabenbezogene Konkretisierung:**

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

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

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



**Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Informatiksysteme und ihr genereller Aufbau</b> (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)	<b>Kapitel 1 „Was macht Informatik“</b> Als Anschauungsmaterial bieten sich Navigationsgeräte an <b>Kapitel 1 Was ist Informatik</b> 1.1 Bereiche der Informatik
<b>2. Der kompetente Umgang mit dem Schulnetzwerk</b> (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		<b>Kapitel 1 „Was macht Informatik“</b> Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule <b>Kapitel 1 Was ist Informatik,</b> Übungen ohne Buch



## **Unterrichtsvorhaben EF-II**

**Thema:** Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

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

### **Vorhabenbezogene Konkretisierung:**

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

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

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 4) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

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



**Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Identifikation von Objekten und Klassen</b> (a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt. (c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.	Die Schülerinnen und Schüler - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).	<b>Kapitel 2 Einführung in die Objektorientierung</b> 2.1 Objektorientierte Modellierung <b>Kapitel 1 Was ist Informatik</b> 1.2 Objektorientierte Programmierung
<b>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</b> (a) Schritte der objektorientierten Analyse, Modellierung und Implementation (b) Analyse und Erprobung der Objekte in Greenfoot-Szenarien		<b>Kapitel 2 Einführung in die Objektorientierung</b> 2.2 Greenfoot-Szenarien Von der Realität zu Objekten Von den Objekten zu Klassen, Klassendokumentation Objekte inspizieren Methoden aufrufen Objektidentität und Objektzustand <b>Kapitel 1 Was ist Informatik</b> 1.3 Erste Schritte in Greenfoot 1.4 Die Modellierung in Greenfoot
<b>3. Implementierung einfacher Aktionen in Greenfoot</b> (a) Quelltext einer Java-Klasse (b) Implementation eigener Methoden, Dokumentation mit JavaDoc (c) Programme übersetzen (Aufgabe des Compilers) und testen		<b>Kapitel 2 Einführung in die Objektorientierung</b> 2.3 Programmierung in Greenfoot Methoden schreiben Programme übersetzen und testen <b>Kapitel 1 Was ist Informatik</b> 1.5 Die Programmierung in Greenfoot





### Unterrichtsvorhaben EF-III

**Thema:** Algorithmische Grundstrukturen in Java

**Leitfragen:** *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

#### **Vorhabenbezogene Konkretisierung:**

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

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

#### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Algorithmen</b> (a) Wiederholungen (While-Schleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methoden aufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),	<b>Kapitel 3 Algorithmen</b> 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 Algorithmen entwickeln <b>Kapitel 2 Algorithmen</b> 2.1 Bedingte Anweisung 2.2 Logische Verknüpfung 2.3 Vorprüfende Wiederholung 2.4 Nachprüfende Wiederholung 2.5 Zählschleife 2.6 Algorithmen entwickeln
<b>2. Variablen und Methoden</b> (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstruktoren	- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	<b>Kapitel 4 Variablen und Methoden</b> 4.1 lokale Variablen 4.2 Methoden 4.3 Attribute <b>Kapitel 4 Methoden und Variablen</b> 3.1 Methoden 3.2 Variablen 3.3 Rückgaben und



(d) Realisierung von Attributen		Parameter
---------------------------------	--	-----------



## Unterrichtsvorhaben EF-IV

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Analoge und digitale Aufbereitung und Verarbeitung von Daten</b> (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - stellen ganze Zahlen und Zeichen in Binärcodes dar (D), - interpretieren Binärcodes als Zahlen und Zeichen (D)	<b>Exkurs „Analog und Digital“</b> <b>2 Kapitel 3 Lineare Datenstrukturen</b> Exkurs: Analog und Digital
<b>2. Der Umgang mit binärer Codierung von Informationen</b> (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners	- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	<b>Exkurs „Binäre Welt“</b> <b>2 Kapitel 2 Konzepte des objektorientierten Modellierens</b> Exkurs: Binäre Welt
<b>3. Aufbau informatischer Systeme</b> (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-		<b>Exkurs „Arbeitsweise eines Computers“</b> Ohne Buch



Prinzips		
----------	--	--

### **Unterrichtsvorhaben EF-V**

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

**Leitfragen:** Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

#### **Vorhabenbezogene Konkretisierung:**

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

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

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

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



**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Umsetzung von Anforderungen in Entwurfsdiagramme</b>                      (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert                      (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</p>	<p>Die Schülerinnen und Schüler                      - analysieren und erläutern eine objektorientierte Modellierung (A),                      - stellen die Kommunikation zwischen Objekten grafisch dar (M),                      - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),                      - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),                      - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),                      - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),                      - modellieren Klassen unter Verwendung von Vererbung (M),                      - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),                      - testen Programme schrittweise anhand von Beispielen (I),                      - interpretieren Fehlermeldungen und korrigieren den Quellcode (I),                      - analysieren und erläutern einfache Algorithmen und Programme (A)                      - modifizieren einfache Algorithmen und Programme (I),                      - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).                      - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),                      - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p><b>Kapitel 6 Klassenentwurf</b>                      6.1. Von der Realität zum Programm                      6.2. Objekte identifizieren                      6.3. Klassen und Beziehungen entwerfen  <b>Kapitel 4 Klassenentwurf</b>                      4.1 Klassen                      4.2 Objekte</p>
<p><b>2. Implementationsdiagramme als erster Schritt der Programmierung</b>                      (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden                      (b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern                      (c) Entwicklung von Klassendokumentationen                      (d) Erstellung von Sequenzdiagrammen als Vorbereitung Vorbereitung für die Programmierung</p>	<p>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),                      - analysieren und erläutern einfache Algorithmen und Programme (A)                      - modifizieren einfache Algorithmen und Programme (I),                      - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).                      - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),                      - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p><b>Kapitel 6 Klassenentwurf</b>                      6.4 Klassen und Beziehungen implementieren                      6.5 Vererbung  <b>2 Kapitel 2 Konzepte des objektorientierten Modellierens</b>                      2.1 Die Welt ist voller Objekte                      2.2 Gut geplant                      -Klassenentwurf                      2.3 Hierarchien machen's einfacher - Vererbung</p>
<p><b>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</b>                      (a) Klassen werden in Java-Quellcode umgesetzt                      (b) Das Geheimnisprinzip wird umgesetzt                      (c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<p>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),                      - analysieren und erläutern einfache Algorithmen und Programme (A)                      - modifizieren einfache Algorithmen und Programme (I),                      - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).                      - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),                      - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p><b>Kapitel 6 Klassenentwurf</b>                      6.4 Klassen und Beziehungen implementieren                      6.5 Vererbung  <b>2 Kapitel 2 Konzepte des objektorientierten Modellierens</b>                      2.2 Gut geplant                      -Klassenentwurf                      2.3 Hierarchien machen's einfacher - Vererbung</p>
<p><b>4. Vererbungsbeziehungen</b>                      (a) Das Grundprinzip der Vererbung wird erarbeitet                      (b) Die Vorteile der Vererbungsbeziehungen                      (c) Vererbung wird implementiert</p>	<p>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),                      - analysieren und erläutern einfache Algorithmen und Programme (A)                      - modifizieren einfache Algorithmen und Programme (I),                      - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).                      - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),                      - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p><b>Kapitel 6 Klassenentwurf</b>                      6.5 Vererbung  <b>2 Kapitel 2 Konzepte des objektorientierten Modellierens</b>                      2.3 Hierarchien machen's einfacher - Vererbung</p>
<p><b>5. Softwareprojekt</b>                      (a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)                      (b) Erweiterung des Spiels um weitere Funktionalitäten                      (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche                      (d) (arbeitsteilige) Implementation des</p>	<p>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),                      - analysieren und erläutern einfache Algorithmen und Programme (A)                      - modifizieren einfache Algorithmen und Programme (I),                      - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).                      - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),                      - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p><b>Kapitel 8 Softwareprojekte</b>                      8.1 Softwareentwicklung                      8.2 Oberflächen  <b>Kapitel 5 Softwareprojekt</b>                      5.1 Softwareentwicklung                      5.2 Dekonstruktion                      5.3 Konstruktion                      5.4 Oberflächen</p>



Spiels		
--------	--	--

### **Unterrichtsvorhaben EF-VI**

**Thema:** Such- und Sortieralgorithmen

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

#### **Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

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

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.



**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Modellierung und Implementation von Datenansammlungen</b>                      (a) Modellierung von Attributen als Felder                      (b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<p>Die Schülerinnen und Schüler                      - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)                      - entwerfen einen weiteren Algorithmus zum Sortieren (M)                      - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)                      - ordnen Attributen lineare Datenansammlungen zu (M)</p>	<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b>                      7.1 Das Feld – Eine Sammlung von Daten  <i>Kapitel 5 Softwareprojekt</i>                      5.5 Felder-Arrays  <i>2 Kapitel 3 Lineare Datenstrukturen</i>                      3.2 Speichern mit Struktur - Arrays</p>
<p><b>2. Explorative Erarbeitung von Suchverfahren</b>                      (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion.                      (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.</p>		<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b>                      Projekteinstieg 1: Suchen                      7.2 Suchen mit System                      Lineare Suche                      Binäre Suche                      Hashing  <i>2 Kapitel 4 Algorithmen</i>                      4.1 Ohne Algorithmen läuft nichts                      4.2 Suchen mit System                      Lineare Suche                      Binäre Suche                      Hashing</p>
<p><b>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</b>                      (a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode                      (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele                      (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche                      (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs                      (e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b>                      Projekteinstieg 2: Sortieren                      7.3 Ordnung ist das halbe Leben!? – Sortieren                      Sortieren                      Selection Sort                      Insertion Sort                      Bubble Sort  <i>2 Kapitel 4 Algorithmen</i>                      4.3 Ordnung ist das halbe Leben!? – Sortieren                      Sortieren                      Selection Sort                      Insertion Sort                      Bubble Sort</p>



## Unterrichtsvorhaben EF-VII

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<p><b>1. Schriftzeichen, Rechenmaschine, Computer</b>                      (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden.                      (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),</li> <li>- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A)</li> <li>- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> </ul>	<p><i>Exkurs „Geschichte der Informatik“</i>  <i>Ohne Buch</i></p>
<p><b>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</b>                      (a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet                      (b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt                      (c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.</p>		<p><i>Exkurs „Informatik und Gesellschaft“</i>  <b>3 Kapitel 2 Informatik und Gesellschaft</b>                      2.2 Datenschutz</p>





## Qualifikationsphase Q1

### 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><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><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> </ul>	<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> </ul>



Freiherr-vom-Stein-Gymnasium Rösrath – Fachbereich Informatik

- Syntax und Semantik einer Programmiersprache - Wirkungen der Automatisierung	- Einzelrechner und Rechnernetzwerke - Grenzen der Automatisierung
---	---



## 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><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><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>	



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



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><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></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: Klassenentwurf – step by step</p> <p><b>2 Kapitel 2 Konzepte des objektorientierten Modellierens</b></p> <p>2.1 Die Welt ist voller Objekte</p> <p>2.2 Gut geplant -Klassenentwurf</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>- 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><li>- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li></ul>	<p>1.3 Gut geplant – Klassenentwurf</p> <p>1.4 Hierarchien machen's einfacher – Vererbung</p> <p><b>2 Kapitel 2 Konzepte des objektorientierten Modellierens</b></p> <p>2.2 Gut geplant -Klassenentwurf</p> <p>2.3 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>- 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> <p><b>3 Kapitel 2 Informatik und Gesellschaft</b></p> <p>2.3 Mensch und Technik – Wer hat die Oberhand</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.



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 Kapitel 3 Lineare Datenstrukturen</b></p> <p>3.2 Speichern mit Struktur - 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>2 Kapitel 3 Lineare Datenstrukturen</b></p> <p>3.3 Wer zuerst kommt,... - Schlagen</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</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>2 Kapitel 3 Lineare Datenstrukturen</b></p> <p>3.4 Daten gut abgelegt -Stapel</p>



<p>Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array</i> (<i>Palindrom</i>))</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 <b>2 Kapitel 3 Lineare Datenstrukturen</b></p> <p>3.5 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 <b>2 Kapitel 3 Lineare Datenstrukturen</b> 3.6 Training</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> <p><b>3 Kapitel 2 Informatik und Gesellschaft, weitere Materialien</b></p>





### **Unterrichtsvorhaben Q1-III**

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

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

#### **Vorhabenbezogene Konkretisierung:**

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

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

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

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.



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 Kapitel 4 Algorithmen</b></p> <p>4.4 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>2 Kapitel 4 Algorithmen</b></p> <p>4.4 Teile die Arbeit – Rekursive Algorithmen</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>2 Kapitel 4 Algorithmen</b></p> <p>4.4 Teile die Arbeit – Rekursive Algorithmen</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> <p><b>3 Kapitel 2 Informatik und Gesellschaft, weitere Materialien</b></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.



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><i>3 Kapitel 4 Endliche Automaten und formale Sprachen</i> Projekteinstieg 2: Treasure Hunt</p> <p>4.2 Automatentheorie</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><i>3 Kapitel 4 Endliche Automaten und formale Sprachen</i> 4.3 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><i>3 Kapitel 4 Endliche Automaten und formale Sprachen</i> 4.5 Training</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><i>3 Kapitel 4 Endliche Automaten und formale Sprachen</i> 4.4 Parser</p> <p><i>3 Kapitel 3 Datenbanken</i> Exkurs: Maschinennahe</p>



		<p>Programmierung</p> <p>Die digitale Welt 100 – Berechenbarkeit</p> <p><b>3 Kapitel 4 Endliche Automaten und formale Sprachen</b></p> <p>4.1 Berechenbarkeit -was kann ein Computer?</p>
--	--	---



## Qualifikationsphase Q2

### 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.



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> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> </ul>	<p><b>Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>5.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume <b>2 Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>Projekteinstieg 1: Spielbäume</p> <p>5.1 Spielen mit Struktur - Baumstrukturen</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>- 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 <b>2 Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>5.2 Zwei Nachfolger sind genug! – Binäre Bäume</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 <b>2 Kapitel 5 Nicht-lineare Datenstrukturen</b></p> <p>5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</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.



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>3 Kapitel 1 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 Kapitel 1 Kommunikation in Netzwerken</b> 1.2 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>3 Kapitel 1 Kommunikation in Netzwerken</b> 1.3 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>3 Kapitel 1 Kommunikation in Netzwerken</b> Exkurs: 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 werden 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>a) Aufbau von Datenbanksystemen und Grundbegriffe</p> <ul style="list-style-type: none"> <li>- Aufgaben und Eigenschaften eines Datenbanksystems</li> <li>- Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema</li> <li>- Problematisierung von Redundanzen, Anomalien und Inkonsistenzen</li> </ul> <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>- Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM,</li> </ul>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</li> <li>- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> <li>- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> </ul>	<p><b>Kapitel 8 Datenbanken</b></p> <p>7.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>7.2 Daten anordnen mit Tabellen</p> <p>Projekteinstieg: Tabellen</p> <p>7.3 Daten filtern mit SQL</p> <p>7.4 Komplexe Filter</p> <p><b>3 Kapitel 3 Datenbanken</b> Exkurs: Kryptologie</p> <p>3.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>3.2 Daten anordnen mit Tabellen</p>



<p>WHERE, JOIN)</p> <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>Projekteinstieg: Tabellen</p> <p>3.3 Operationen auf Tabellen</p> <p>3.4 Komplexe Filter</p>
<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 Kapitel 3 Datenbanken</b></p> <p>3.5 Datenbankentwurf</p> <p>3.6 Umsetzung des ER-Modells</p> <p>Wiederaufgriff des Projekteinstiegs</p> <p>3.7 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> <p><b>3 Kapitel 3 Datenbanken</b></p> <p>3.8 Training</p>



### **3. Grundsätze der Leistungsbewertung und Leistungsrückmeldung**

Bei der Leistungsbewertung werden alle im Unterricht geförderten Kompetenzen in die Bewertung einbezogen. Diese können auf verschiedene Art und Weise deutlich werden: in schriftlichen Leistungen, mündlichen Beiträgen, bei der Umsetzung praktischer Programmieraufgaben oder in größeren Projekten. Bewertet werden dabei die richtige Anwendung des Fachwissens, aber auch die Art der Darstellung. Dazu gehört die richtige Verwendung der Fachsprache ebenso wie die Anwendung von fachlichen Darstellungsformen und die Klarheit der Darstellung.

Die Anforderungen werden den Schülerinnen und Schülern transparent gemacht.

#### **A. Schriftlicher Anteil.**

In allen Jahrgangsstufen, in denen Informatik unterrichtet wird, werden eine (Stufe EF) oder zwei (alle anderen Stufen) schriftliche Arbeiten bzw. Klausuren geschrieben. Der Durchschnitt dieser schriftlichen Ergebnisse bildet den schriftlichen Anteil der Leistungsbewertung und geht etwa zur Hälfte in die Gesamtnote ein.

Bei der Bildung der Noten der Klausuren wird ein Punkteschema verwendet, dass sich zunehmend an der für das Abitur vorgesehenen Punkteverteilung orientiert.

#### **B. Sonstige Mitarbeit.**

Die zweite Hälfte der Gesamtnote wird durch die sonstige Mitarbeit gebildet. Dazu gehören:

- Mitarbeit im Unterricht.
- Präsentation von Arbeitsergebnissen
- Mitarbeit in Partner- und Gruppenarbeitsphasen
- Referate, Präsentationen, Vorträge
- Praktische Leistungen am Computer
- Schriftliche Leistungen wie z.B. Dokumentationen zu Programmen, Protokolle, Zusammenfassungen von Inhalten, Lernerfolgskontrollen