

# **Schulinterner Lehrplan des Franz-Stock Gymnasiums Arnsberg**

**zum Kernlehrplan für die gymnasiale Oberstufe**

## **Informatik**

**(Stand: 20.01.2017)**

# 1 Die Fachgruppe Informatik des Franz-Stock Gymnasiums Arnsberg

Beim Franz-Stock Gymnasium handelt es sich um eine Schule mit zurzeit ca. 1000 Schülerinnen und Schülern. Das Einzugsgebiet der Schule umfasst den Stadtteil Neheim-Hüsten, das weitere Stadtgebiet sowie die umliegenden Dörfer und Gemeinden.

Das Fach Informatik wird am Franz-Stock Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweistündig im Rahmen eines kombinierten Ph/If/Tk Kurses unterrichtet und von ca. einem Viertel der Schülerinnen und Schüler besucht. Der Unterricht erfolgt dabei in enger Verzahnung mit Inhalten der Mathematik, Technik und Informatik.

In Angebotsform existieren darüber hinaus:

1. Scratch in Klasse 7 (Ganztagsangebot)
2. Informatik & Technik ab Klasse 5 (Ganztagsangebots)

In der Sekundarstufe II bietet das Franz-Stock Gymnasium für die Schülerinnen und Schüler in allen Jahrgangsstufen jeweils ein bis zwei Grundkurse im Fach Informatik an.

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

Der Unterricht der Sekundarstufe II wird mithilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei zusätzlich eine didaktische Bibliothek zum Einsatz, welche das Erstellen von grafischen Programmen erleichtert. Eine Konkretisierung findet in den Unterrichtsvorhaben statt.

Durch projektartiges Vorgehen und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

Der Austausch von Materialien und Unterrichtsvorhaben, die Diskussion von Lehr- und Lernprozessen sowie die Reflexion und eventuelle Modifikation des schulinternen Curriculums durch die Mitglieder der Fachkonferenz Informatik stellt einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Es existieren zwei Computerräume mit je ca. 15 Computerarbeitsplätzen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen.

Der Unterricht erfolgt in Unterrichtseinheiten von 60, 75 oder 90 Minuten.

Die Angabe des Zeitbedarfs für die Unterrichtsvorhaben geht der Vergleichbarkeit halber weiter von Unterrichtsstunden von 45 Minuten aus.

## 2 Entscheidungen zum Unterricht

### 2.1 Unterrichtsvorhaben

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

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

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Die Themen an den gemeinsamen Rändern der Q1 und Q2 können ggf. der jeweils anderen Jahrgangsstufe zugeordnet werden. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 65 Prozent der Bruttounterrichtszeit verplant.

Während die Aussagen zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (s. unten) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule.

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

Bei den Konkretisierungen der Unterrichtsvorhaben wird Bezug genommen auf das Unterrichtswerk:

Informatik – Einführungsphase  
Schülerband 1  
Schöningh ISBN 978-3-14-037126-1

Sowie das zum 31.07.2015 erscheinen sollende Unterrichtswerk:

NEU: Informatik - Lehrwerk für die gymnasiale Oberstufe - Neubearbeitung  
Schülerband 2  
Qualifikationsphase  
ISBN: 978-3-14-037127-8

## 2.1.1 Übersichtsraster Unterrichtsvorhaben und Konkretisierungen

Dieser Lehrplan hat den Anspruch, alle im Kernlehrplan geforderten Kompetenzen abzubilden.

Die Umsetzung des Lehrplans folgt pro Unterrichtsvorhaben jeweils auf einer Übersichts- und einer dazugehörigen Konkretisierungsebene. Die Übersichtsebene ist für die Kolleginnen und Kollegen verpflichtend. Die hier dargestellten Konkretisierungen beziehen sich auf das bereits genannte Schulbuch und können anhand der Lehrtexte, Projekte und Aufgaben umfassend umgesetzt werden.

### Didaktische Lernumgebung

Die Unterrichtsvorhaben in Informatik hängen in ihrer konkreten Umsetzung von der gewählten Programmierumgebung ab.

In diesem Lehrplan wird zur Einführung in die objektorientierte Programmierung die didaktische Lernumgebung Greenfoot gewählt. Die Lehrtexte und Aufgaben des Lehrwerks „Informatik 1“ beziehen sich auch speziell auf ein bestimmtes Greenfoot-Szenario.

Nach der Einarbeitungsphase sollte auf eine komplexere Lernumgebung umgestellt werden. Hier bieten sich verschiedene Möglichkeiten, z.B.: BlueJ, Eclipse, NetBeans etc. an.

## 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>Zeitbedarf:</b> 5 Stunden</p>	<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>Zeitbedarf:</b> 10 Stunden</p>
<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><li>- Syntax und Semantik einer Programmiersprache</li><li>- Analyse, Entwurf und Implementierung einfacher Algorithmen</li></ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>	<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><li>- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten</li></ul> <p><b>Zeitbedarf:</b> 5 Stunden</p>

<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>Zeitbedarf:</b> 20 Stunden</p>	<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>Zeitbedarf:</b> 9 Stunden</p>
<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> <li>- Dateisystem</li> </ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>	<p><b>Summe Einführungsphase:</b> <b>77 Stunden</b></p>

## 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 Informatiksystems 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 an diesem Informatiksystem erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Systeme 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.

**Zeitbedarf:** 5 Stunden

### **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 Informatiksysteme aus der Lebenswelt der SchülerInnen an
<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> Schulnetzwerk, Benutzer- und Datenschutzbestimmungen der Schule

## **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.

**Zeitbedarf:** 10 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Identifikation von Objekten und Klassen</b></p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M),</li> <li>- stellen den Zustand eines Objekts dar (D),</li> <li>- modellieren Klassen mit ihren Attributen und ihren Methoden (M),</li> <li>- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>- implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.1 Objektorientierte Modellierung</p>
<p><b>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</b></p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>		<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“</p> <p>Von der Realität zu Objekten</p> <p>Von den Objekten zu Klassen, Klassendokumentation</p> <p>Objekte inspizieren</p> <p>Methoden aufrufen</p> <p>Objektidentität und Objektzustand</p>
<p><b>3. Implementierung einfacher Aktionen in Greenfoot</b></p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.3 Programmierung in Greenfoot</p> <p>Methoden schreiben</p> <p>Programme übersetzen und testen</p>

## 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.

**Zeitbedarf:** 20 Stunden

### **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 Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	<b>Kapitel 3 Algorithmen</b> 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 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 (d) Realisierung von Attributen		<b>Kapitel 4 Variablen und Methoden</b> 4.1 lokale Variablen 4.2 Methoden 4.3 Attribute

## 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.

**Zeitbedarf:** 5 Stunden

### **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) - beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A)	<i>Exkurs „Analog und Digital“</i>
<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	- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	<i>Exkurs „Binäre Welt“</i>
<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-Prinzips		<i>Exkurs „Arbeitsweise eines Computers“</i>

## **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.

**Zeitbedarf:** 20 Stunden

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

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</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern eine objektorientierte Modellierung (A),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>- modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>- testen Programme schrittweise anhand von Beispielen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- analysieren und erläutern einfache Algorithmen und Programme (A)</li> </ul>	<p><b>Kapitel 6 Klassenentwurf</b>            6.1. Von der Realität zum Programm            6.2. Objekte identifizieren            6.3. Klassen und Beziehungen entwerfen</p>
<p><b>2. Implementationsdiagramme als erster Schritt der Programmierung</b>            (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und Getter und Setter            (b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern            (c) Entwicklung von Klassendokumentationen            (d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</p>	<ul style="list-style-type: none"> <li>- modifizieren einfache Algorithmen und Programme (I),</li> <li>- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).</li> <li>- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</li> </ul>	<p><b>Kapitel 6 Klassenentwurf</b>            6.4 Klassen und Beziehungen implementieren            6.5 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><b>Kapitel 6 Klassenentwurf</b>            6.4 Klassen und Beziehungen implementieren            6.5 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><b>Kapitel 6 Klassenentwurf</b>            6.5 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 Spiels</p>		<p><b>Kapitel 7 Softwareprojekte</b>            7.1 Softwareentwicklung            7.2 Oberflächen</p>

## **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.

**Zeitbedarf:** 9 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Modellierung und Implementation von Datenansammlungen</b></p> <p>(a) Modellierung von Attributen als Felder</p> <p>(b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)</li> <li>- entwerfen einen weiteren Algorithmus zum Sortieren (M)</li> <li>- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)</li> <li>- ordnen Attributen lineare Datenansammlungen zu (M)</li> </ul>	<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>7.1 Das Feld – Eine Sammlung von Daten</p>
<p><b>2. Explorative Erarbeitung von Suchverfahren</b></p> <p>(a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion.</p> <p>(b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.</p>		<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>Projekteinstieg 1: Suchen</p> <p>7.2 Suchen mit System</p> <p>Lineare Suche</p> <p>Binäre Suche</p> <p>Hashing</p>
<p><b>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>(e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>Projekteinstieg 2: Sortieren</p> <p>7.3 Ordnung ist das halbe Leben!?! – Sortieren</p> <p>Sortieren</p> <p>Selection Sort</p> <p>Insertion Sort</p> <p>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 wird 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.

**Zeitbedarf:** 8 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<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.	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), - erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)	<i>Exkurs „Geschichte der Informatik“</i>  <i>ggf. durch Unterrichtsgang in HNF ersetzbar</i>
<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.		<i>Exkurs „Informatik und Gesellschaft“</i>  <i>ggf. durch Unterrichtsgang in HNF ersetzbar</i>

## 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> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Wirkung der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 10 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q1-II</u></b></p> <p><b>Thema:</b> Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> </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> </ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>
<p><b><u>Unterrichtsvorhaben Q1-III</u></b></p> <p><b>Thema:</b> Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>	<p><b><u>Unterrichtsvorhaben Q1-IV</u></b></p> <p><b>Thema:</b> Automaten und formale Sprachen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Formale Sprachen und Automaten</li> <li>- Informatiksysteme</li> </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> <p><b>Zeitbedarf:</b> 20 Stunden</p>
<p><b>Summe Qualifikationsphase 1: 70 Stunden</b></p>	

## Qualifikationsphase – Q2

### Unterrichtsvorhaben Q2-I

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

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 20 Stunden

### Unterrichtsvorhaben Q2-II

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

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen, Wirkungen der Automatisierung

**Zeitbedarf:** 16 Stunden

### Unterrichtsvorhaben Q2-III

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

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 20 Stunden

**Summe Qualifikationsphase 2: 56 Stunden**

## **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.

**Zeitbedarf:** 14 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</b></p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- 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>2.1 Die Welt ist voller Objekte  ProjektEinstieg: Klassenentwurf – step by step</p>
<p><b>2. Erweiterung der objektorientierten Programmierung</b></p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> <li>- 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> <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> </ul>	<p>2.2 Gut geplant – Klassenentwurf  2.3 Vererbungshierarchien nutzen</p>
<p><b>3. Mensch und Technik</b></p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> <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>Die digitale Welt 001 - Mensch und Technik</p>
<p><b>4. Übung und Vertiefung der OOM / OOP</b></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>Prüfungsvorbereitung</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 einigen Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

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

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

**Zeitbedarf:** 20 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Die Datenstruktur <b>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 und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A)</li> <li>- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M)</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> </ul>	<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Anforderungen an eine Datenstruktur</li> <li>- Speichern mit Struktur</li> </ul> <p><i>neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)</i></p>
<p>2. Die Datenstruktur <b>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>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I)</li> <li>- testen und korrigieren Computerprogramme (I)</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D)</li> <li>- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D)</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> </ul>	<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Wer zuerst kommt</li> <li>- Objekte miteinander verketteten</li> <li>- Verwaltung und Inhalt</li> <li>- Funktionen der Queue</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist.</li> <li>- Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus dem Verkehr gewunken. Es werden so lange Fahrzeuge kontrolliert, bis eine gewissen Menge an Verstößen vorliegt oder Autos kontrolliert wurden.</li> </ul>
<p>3. Die Datenstruktur <b>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></p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer</p>	<ul style="list-style-type: none"> <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> </ul>	<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Daten gut abgelegt – Stapel</li> <li>- Funktionen der Datenstruktur Stapel</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Standardoperationen/Basiskompetenz (Stapel Münzen/ CDs) zur Umsetzung der gegebenen Funktionen der Klasse <i>Stack</i></li> <li>- Biber und Teller</li> </ul>

<p>Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>		<p>Es gibt große und kleine Biber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl <i>Queue</i> als auch <i>Stack</i> verwendet werden.</p> <ul style="list-style-type: none"> <li>- Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.</li> </ul>
<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><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Flexibel für alle Fälle – (die) lineare Liste</li> <li>- Funktionen der Datenstruktur Liste</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- LEDs</li> <li>- Textzeilen verarbeiten</li> </ul>
<p><b>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</b></p>		<p>Prüfungsvorbereitung</p>
		<p><i>Projekteinstieg Heldenspiel</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>

## **Unterrichtsvorhaben Q1-III**

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

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

**Zeitbedarf:** 20 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Eigenschaften von Algorithmen</b></p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> </ul>	<p>4.1 Ohne Algorithmen läuft nichts</p> <p>4.2 Teile die Arbeit – rekursive Algorithmen</p>
<p><b>2. Suchen in Listen und Arrays</b></p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I).</li> <li>- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> </ul>	<p>4.3 Suchen – iterativ und rekursiv</p>
<p><b>3. Sortieren auf Listen und Arrays</b></p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</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>- testen Programme systematisch anhand von Beispielen (I),</li> </ul>	<p>4.4 Sortieren – iterativ und rekursiv</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?

**Vorhabensbezogene Konkretisierung:**

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert. An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht.

**Zeitbedarf:** 20 Std.

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Endliche Automaten</b></p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEA 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>- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D)</li> <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 endlicher 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> </ul>	<p>Vom realen Automaten zum Modell</p> <p>Projekteinstieg: Schatzsuche</p> <p>Der Mealy-Automat</p> <p>Der erkennende endliche Automat</p> <p>Wort und Sprache</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 auf</li> </ul>	<p>Grammatiken regulärer Sprachen</p>
<p><b>3. Übungen und Vertiefungen</b></p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>		<p>Prüfungsvorbereitung</p>
<p><i>Projekteinstieg</i></p> <p><i>Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben</i></p>		

## **Unterrichtsvorhaben Q2-I**

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

Möchte man Daten geordnet speichern, so 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 Item 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 Item modelliert und (ggf. in Teilen) implementiert.

**Zeitbedarf:** 20 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Aufbau von Baumstrukturen und Grundbegriffe</b></p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> </ul>	<p>6.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p><b>2. Binäre Bäume</b></p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- 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>6.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p><b>3. Binäre Suchbäume</b></p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> </ul>	<p>6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>		<p>Prüfungsvorbereitung</p>

## **Unterrichtsvorhaben Q2-II**

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

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

**Vorhabenbezogene Konkretisierung:**

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

**Zeitbedarf:** 12 Stunden

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</b></p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> <li>- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D),</li> <li>- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).</li> </ul>	<p>7.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p><b>2. Aufbau von Netzwerken und Kommunikationsregeln</b></p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen</p>		<p>7.2 Ohne Protokoll läuft nichts - Netzwerke</p>
<p><b>3. Aufgabenteilung in Netzwerken durch Server und Client</b></p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>(b) Protokolle zwischen Client und Server</p>		<p>7.3 Einer für alle – Client-Server-Struktur</p>
<p><b>4. Kryptologie</b></p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 100 – Kryptologie</p>
<p><b>5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</b></p>		<p>Prüfungsvorbereitung</p>

## **Unterrichtsvorhaben Q2-III**

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

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

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

**Zeitbedarf:** 20 Stunden

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

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, WHERE, JOIN)</li> <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>	<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>- analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>- erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>- bestimmen Primär- und Sekundärschlüssel (M),</li> <li>- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>- modifizieren eine Datenbankmodellierung (M),</li> <li>- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>- überführen Datenbankschemata in vorgegebene Normalformen (M),</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>- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li> </ul>	<p><b>Kapitel 8 Datenbanken</b></p> <p>8.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>8.2 Daten anordnen mit Tabellen</p> <p>Beispiel: Buchhandlung</p> <p>Redundanzen, Anomalien und Inkonsistenzen</p> <p>Projekteinstieg: Tabellen</p> <p>8.3 Daten filtern mit SQL</p> <p>8.4 Komplexe Filter</p> <p>Aufgaben</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</p>		<p>8.5 Datenbankentwurf</p> <p><i>Beispiel: Online-Buchhandel</i></p> <p><i>Datenanalyse und Entwurf</i></p> <p>8.6 Umsetzung des ER-Modells</p> <p><i>Entitätsmengen</i></p> <p><i>m:n-Beziehungen</i></p> <p><i>1:n-Beziehungen</i></p> <p><i>1:1-Beziehungen</i></p>

<p>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><i>Wiederaufgriff des Projekteinstiegs</i></p> <p><i>8.7 Datenbanken verbessern durch Normalformen</i></p>
<p><b>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b></p>		<p>Prüfungsvorbereitung</p>

## 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Franz-Stock Gymnasiums die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz. Generell sind die Bereiche der sonstigen Mitarbeit und Klausuren etwa gleichwertig zu berücksichtigen.

### 2.3.1 Beurteilungsbereich Klausuren

#### Verbindliche Absprachen:

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

#### Instrumente:

- (a) Einführungsphase: 1 Klausur je Halbjahr  
Dauer der Klausur: 2 Unterrichtsstunden
- (b) Grundkurse Q 1: 2 Klausuren je Halbjahr  
Dauer der Klausuren: 2 Unterrichtsstunden
- (c) Grundkurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 3 Unterrichtsstunden
- (d) Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- (e) Anstelle einer Klausur kann in Q1.2 eine Facharbeit geschrieben werden.

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

#### Kriterien

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt in der Q1/Q2 über ein Raster mit Punkten (analog gem. § 48 Abs. 3 SchulG in der EF):

Note	Punkte	Punktzahl in Prozent
sehr gut plus	15	95-100
sehr gut	14	90-94
sehr gut minus	13	85-89
gut plus	12	80-84
gut	11	75-79
gut minus	10	70-74
befriedigend plus	9	65-69
befriedigend	8	60-64
befriedigend minus	7	55-59
ausreichend plus	6	50-54
ausreichen	5	45-49
ausreichen minus	4	39-44
mangelhaft plus	3	33-38
mangelhaft	2	27-32
mangelhaft minus	1	20-26
ungenügend	0	0-19

(Analog Abiturverfügung 2013)

Diese Notenzuordnung ist als Empfehlung zu betrachten, Abweichungen sind aus pädagogischen Gründen möglich. Insbesondere sind bei Clusterbildungen über die Grenzen hinweg durchaus Korrekturen möglich.

## 2.3.2 Beurteilungsbereich Sonstige Mitarbeit

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn des Schuljahres genannt.

### Leistungsaspekte

#### Mündliche Leistungen

- (a) Beteiligung am Unterrichtsgespräch
- (b) Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- (c) Präsentation von Arbeitsergebnissen
- (d) Mitarbeit in Partner-/Gruppenarbeitsphasen

#### Praktische Leistungen am Computer

- (e) Implementierung, Test und Anwendung von Informatiksystemen

#### Sonstige schriftliche Leistungen

- (f) Schriftliche Übung im Rahmen der gesetzlichen Regelungen in Verantwortung des Fachlehrers
- (g) Bearbeitung von schriftlichen Aufgaben im Unterricht

### Kriterien

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

Die Bewertungskriterien stützen sich auf

- (a) die Qualität der Beiträge,
- (b) die Quantität der Beiträge und
- (c) die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- (d) die sachliche Richtigkeit,
- (e) die angemessene Verwendung der Fachsprache,
- (f) die Darstellungskompetenz,
- (g) die Komplexität und den Grad der Abstraktion,
- (h) die Selbstständigkeit im Arbeitsprozess,
- (i) die Präzision und
- (j) die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- (k) das Einbringen in die Arbeit der Gruppe,
- (l) die Durchführung fachlicher Arbeitsanteile und
- (m) die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- (n) die Dokumentation des Arbeitsprozesses,
- (o) den Grad der Selbstständigkeit,
- (p) die Reflexion des eigenen Handelns und
- (q) die Aufnahme von Beratung durch die Lehrkraft.

## **Grundsätze der Leistungsrückmeldung und Beratung**

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres im Unterricht transparent gemacht. Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Elternsprechtagen.

## **3 Entscheidungen zu Exkursionen und Facharbeit**

### **Vorbereitung auf die Erstellung der Facharbeit**

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

### **Exkursionen**

In der Einführungsphase kann im Rahmen des Unterrichtsvorhabens „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum Heinz Nixdorf MuseumsForum durchgeführt werden.

## **4 Qualitätssicherung und Evaluation**

Dieses schulinterne Curriculum ist nach Erlass des Kernlehrplanes verbindlich. Änderungen für den nächsten Durchgang der Einführungsphase oder Qualifikationsphase müssen ggf. beschlossen werden, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.