

**Pestalozzi – Gymnasium Herne**

**Schulinterner Lehrplan  
Informatik**

## Einführungsphase – Grundkurs

Im Folgenden sollen Unterrichtsvorhaben für das Fach Informatik dargestellt werden. Alle hier aufgeführten Vorhaben beziehen sich auf Grundkurse in der Einführungsphase.

Zu jedem Unterrichtsvorhaben ist eine Anknüpfung an den Kernlehrplan Informatik in Form von Kompetenzbezügen gegeben. Die aufgeführten Kompetenzen sind dabei so zu verstehen, dass das entsprechende Unterrichtsvorhaben zum Erwerb derselben beiträgt.

Kompetenzerwerb ist ein kontinuierlicher und kumulativer Prozess, der sich über längere Zeiträume hinzieht und die wiederholte Beschäftigung mit entsprechenden fachlichen Gegenständen und Themen in variierenden Anwendungssituationen oder auf zunehmenden Anforderungsniveaus voraussetzt. Es kann daher nicht der Anspruch erhoben werden, dass die aufgeführten Kompetenzen nach Abschluss lediglich eines Unterrichtsvorhabens vollständig erworben wurden.

<p>Unterrichtsvorhaben E-I Thema: Überblick über das Fach Informatik, Grundlagen von Informatiksystem und Geschichte der Informatik</p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Digitalisierung Einzelrechner Dateisysteme Internet Einsatz von Informatiksystemen Wirkungen der Automatisierung Geschichte der automatischen Datenverarbeitung</p> <p>Zeitbedarf: 6 Stunden</p>	<p>Unterrichtsvorhaben E-II Thema: Grundlagen der Programmierung mit Java und einfache Algorithmik</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 33 Stunden</p>
<p>Unterrichtsvorhaben E-III Thema: Entwurf und Implementierung einfacher Sortierverfahren und eines klassischen kryptographischen Verfahrens</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen</p>	<p>Unterrichtsvorhaben E-IV Thema: Objektorientierte Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen, Simulationen und grafischen Oberflächen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung</p>

<p>Unterrichtsvorhaben E-I Thema: Überblick über das Fach Informatik, Grundlagen von Informatiksystem und Geschichte der Informatik</p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Digitalisierung Einzelrechner Dateisysteme Internet Einsatz von Informatiksystemen Wirkungen der Automatisierung Geschichte der automatischen Datenverarbeitung</p> <p>Zeitbedarf: 6 Stunden</p>	<p>Unterrichtsvorhaben E-II Thema: Grundlagen der Programmierung mit Java und einfache Algorithmik</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 33 Stunden</p>
<p>Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache Algorithmen zum Suchen und Sortieren</p> <p>Zeitbedarf: 12 Stunden</p>	<p>Algorithmen Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 36 Stunden</p>

## Qualifikationsphase – Grundkurs

Qualifikationsphase I	
<p>Unterrichtsvorhaben Q1-I Thema: Rekursive Algorithmen und Backtracking in Anwendungskontexten</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Algorithmen Formale Sprachen und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Grenzen der Automatisierung</p> <p>Zeitbedarf: 20 Stunden</p>	<p>Unterrichtsvorhaben Q1-II Thema: Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen</p> <p>Zeitbedarf: 25 Stunden</p>
<p>Unterrichtsvorhaben Q1-III Thema: Modellierung und Implementierung dynamische nichtlineare Datenstrukturen am Beispiel der Binärbäume</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen</p>	<p>Unterrichtsvorhaben Q1-IV Thema: Sicherheit und Datenschutz in Netzstrukturen</p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Einzelrechner und Rechnernetzwerke Sicherheit Nutzung von Informatiksystemen, Wirkungen der Automatisierung</p> <p>Zeitbedarf: 15 Stunden</p>

Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen  Zeitbedarf: 20 Stunden	
Summe Qualifikationsphase I: 80 Stunden	

Qualifikationsphase II	
<p>Unterrichtsvorhaben Q2-I Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprache und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Sicherheit Wirkung der Automatisierung</p> <p>Zeitbedarf: 25 Stunden</p>	<p>Unterrichtsvorhaben Q2-II Thema: Endliche Automaten und Formale Sprachen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Endliche Automaten Grammatiken regulärer Sprachen Möglichkeiten und Grenzen von Automaten und formalen Sprachen Nutzung von Informatiksystemen</p> <p>Zeitbedarf: 25 Stunden</p>
<p>Unterrichtsvorhaben Q2-III Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</p> <p>Zentrale Kompetenzen: Argumentieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Einzelrechner und Rechnernetzwerke Grenzen der Automatisierung</p> <p>Zeitbedarf: 10 Stunden</p>	
Summe Qualifikationsphase II: 60 Stunden	

## Unterrichtsvorhaben Nr. 1

Thema: Überblick über das Fach Informatik, Grundlagen von Informatiksystem und Geschichte der Informatik

*Leitfragen:* Mit welchen Themen befasst sich das Fach Informatik in der Schule? Wie funktioniert ein moderner Computer? Welche Entwicklung durchlief die moderne Datenverarbeitung?

*Zeitbedarf:* 6 Stunden

### *Absprachen zur vorhabenbezogenen Konkretisierung:*

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Da einige Schülerinnen und Schüler das Fach zum ersten Mal in der Einführungsphase belegen, wird zu Beginn ein Überblick über die Themen des Schulfachs Informatik gegeben. Unter anderem wird auf den zentralen Begriff der Information eingegangen und die Möglichkeit der Codierung von Daten, insbesondere wird die Binärdarstellung von Zahlen thematisiert. Stationen der geschichtlichen Entwicklung werden angesprochen wie z.B. Speichermedien, von-Neumann-Architektur und das EVA-Prinzip. Außerdem werden die SuS in die konkrete Nutzung der Informatiksysteme an der Schule eingewiesen.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
Allgemeine Einführung <ul style="list-style-type: none"><li>• Übersicht über das Fach</li><li>• Einführung in die Informatiksysteme der Schule</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• nutzen die im Unterricht eingesetzten Informatiksysteme</li><li>• selbstständig, sicher, zielführend und verantwortungsbewusst</li><li>• (D),</li><li>• nutzen das verfügbare Informatiksystem zu strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K),</li><li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li></ul>	Computerraum der Schule
Grundlagen von Informatiksystemen und Überblick über die Geschichte der Informatik <ul style="list-style-type: none"><li>• Darstellung von Zahlen im Binärsystem</li><li>• Von-Neumann-Architektur</li><li>• Geschichte der Datenverarbeitung</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li><li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li><li>• beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am</li><li>• Beispiel der „VonNeumann-Architektur“ (A).</li><li>• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A)</li></ul>	

## Unterrichtsvorhaben Nr. 2

*Thema:* Grundlagen der Programmierung mit Java und einfache Algorithmen

*Leitfragen:* Aus welchen Bausteinen besteht ein Java-Programm? Welche Datentypen und Kontrollstrukturen stehen in Java zur Verfügung und wie nutzt man diese? Wie verläuft der Entwicklungsprozess eines Java-Programms?

*Zeitbedarf:* 33 Stunden

### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Die Grundlagen zur Programmieretechnik mit Java werden durch die SuS selbstständig und individuell erarbeitet. Dazu zählen: einfache Datentypen, Variablenkonzept, Wertzuweisungen, Kontrollstrukturen, Methodenkonzept, Arrays. Die Schülerinnen und Schüler wenden ihr Wissen kontextbezogen auf kleinere Problemstellungen an. Dabei durchlaufen sie den kontinuierlichen Prozess von Implementieren, Compilieren und Testen. Die Schülerinnen und Schüler dokumentieren kontinuierlich und selbstständig ihren Lernfortschritt in einem Lerntagebuch.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
Einführung in die Programmierung <ul style="list-style-type: none"><li>• Kap. 1: Erste Schritte</li><li>• Kap. 2: Programmaufbau</li><li>• Kap. 3: Datentypen und Operationen</li><li>• Kap. 4: Variablen</li><li>• Kap. 5: Verzweigungen</li><li>• Kap. 6: Schleifen</li><li>• Kap. 7: Arrays</li><li>• Kap. 8: Methoden</li></ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li><li>• modifizieren einfache Algorithmen und Programme (I),</li><li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li><li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li><li>• testen Programme schrittweise anhand von Beispielen (I).</li><li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li></ul>	<ul style="list-style-type: none"><li>• BlueJ</li><li>• Structorizer</li><li>• Lerntagebuch</li></ul>



### Unterrichtsvorhaben Nr. 3

**Thema:** Entwurf und Implementierung einfacher Sortierverfahren und eines klassischen kryptologischen Verfahrens,.

**Leitfragen:** Wie kann man die bisherigen Erkenntnisse im Umgang mit Arrays zum Sortieren von großen Datenmengen und zur einfachen „Verschlüsselung“ von Daten benutzen? Wie effizient ist ein Algorithmus hinsichtlich seines Laufzeitverhaltens?

**Zeitbedarf:** 12 Stunden

#### **Absprachen zur vorhaben-bezogene Konkretisierung:**

Aufbauend auf den programmiertechnischen Grundlagen des letzten Unterrichtsvorhabens können die SuS nun ihr Wissen zu Arrays und Strings kontextbezogen anwenden: Dazu werden mindestens zwei der einfachen Sortierverfahren behandelt, indem sie simuliert, implementiert und getestet werden. In der anschließenden Analyse reflektieren die SuS das Laufzeitverhalten der Algorithmen im best, worst und normal case. Darüber hinaus wird ein einfaches kryptologisches Verfahren (z.B. CAESAR) behandelt. Neben der Durchführung der einfachen Verschlüsselungstechnik werden die zur Implementierung nötigen String-Operationen vertieft. Hierzu gehört auch die Thematisierung des ASCII-Codes. Die Schülerinnen und Schüler reflektieren die Sicherheitsaspekte des Verfahrens.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
Implementierung und Analyse einfacher Sortier- und Suchverfahren auf Arrays	Die Schülerinnen und Schüler <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 Zeitaufwand und Speicherplatzbedarf (A).</li></ul>	<ul style="list-style-type: none"><li>• JavaEditor</li><li>• BlueJ</li><li>• MathePrisma Uni Wuppertal</li></ul>
Implementierung und Analyse eines einfachen kryptologischen Verfahrens	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li><li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li><li>• testen Programme schrittweise anhand von Beispielen (I)</li></ul>	<ul style="list-style-type: none"><li>• MathePrisma Uni Wuppertal</li><li>• BlueJ</li></ul>

## Unterrichtsvorhaben Nr. 4

**Thema:** Objektorientierte Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen, Simulationen und grafischen Oberflächen .

**Leitfragen:** Was sind Objekte, was sind Klassen? Wie programmiert man objektorientiert? Was bedeutet Vererbung? Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche realisieren? Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?

**Zeitbedarf:** 36 Stunden

**Absprachen zur vorhaben-bezogene Konkretisierung:**

Die Unterrichtssequenz orientiert sich an dem Buch Bernhard Schriek: Informatik mit Java

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Einführung in die OO</p> <ul style="list-style-type: none"><li>• Was ist ein Objekt? Was ist eine Klasse?</li><li>• Methoden und Attribute</li><li>• Objekte erzeugen</li><li>• Objektbeziehungen und Objektinteraktionen</li><li>• Geheimnisprinzip</li><li>• Vererbung</li></ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"><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>• modellieren Klassen unter Verwendung von Vererbung (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>• stellen den Zustand eines Objekts dar (D),</li><li>• stellen die Kommunikation</li><li>• zwischen Objekten 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><li>• analysieren und erläutern eine</li><li>• objektorientierte Modellierung (A),</li><li>• implementieren Klassen in einer</li><li>• Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li></ul>	<p>Bernhard Schriek: Informatik mit Java Bd. I</p>

## Qualifikationsphase – Grundkurs

### *Unterrichtsvorhaben Q1-I*

*Thema:* Rekursive Algorithmen und Backtracking in Anwendungskontexten

*Leitfragen:* Wie können komplexe, rekursiv definierte Probleme informatisch gelöst werden?

Gibt es schnelle (rekursiv definierte) Sortier- und Suchverfahren?

*Zeitbedarf:* 20 Stunden

#### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Ausgehend vom einem Problem wie z. B. "Türme von Hanoi" wird Rekursion als fundamentale Idee der Informatik zunächst im mathematischen, danach aber auch im informatischen Zusammenhang angewendet. Dabei wird zwischen linearen und verzweigten Rekursionen unterschieden und das Laufzeitverhalten bei hoher Rekursionstiefe analysiert. Verschiedene NP-vollständige Probleme (wie z. B. Rucksack, n-Damen, Springer, Irrgarten, etc.) werden algorithmisch rekursiv formuliert. Bereits bekannte Such- und Sortierverfahren (z. B. Sortieren durch Einfügen, Sortieren durch Auswahl, Sequentielle Suche) werden rekursiv formuliert und durch leistungsfähigere Verfahren (z. B. Quicksort, Mergesort, Heapsort, Binäre Suche) ergänzt. Die neuen Verfahren werden implementiert.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Entwicklung der Rekursion als fundamentale Idee der Informatik</p> <ul style="list-style-type: none"> <li>• rekursive Formeln</li> <li>• rekursive Funktionen / Methoden</li> <li>• rekursive Programmierung</li> </ul>	<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> </ul>	<ul style="list-style-type: none"> <li>• Türme von Hanoi mit Schwerpunkt auf Zahl der Versetzungsoperationen</li> <li>• Protokollierung der Versetzungen</li> </ul>
<p>Rekursion in mathematischen und informatischen Kontexten</p> <ul style="list-style-type: none"> <li>• Rekursion in mathematischen Kontexten</li> <li>• Analyse und Darstellung des rekursiven Ablaufs einer Methode</li> <li>• Analyse des Laufzeitverhaltens linearer und verzweigter Rekursion</li> </ul>	<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>• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen(I).</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<ul style="list-style-type: none"> <li>• Fakultätsfunktion (lineare Rekursion)</li> <li>• Fibonacci-Funktion (verzweigte Rekursion)</li> <li>• ggT (verzweigte Rekursion)</li> <li>• evtl. Fraktale (Kochkurve, Sierpinski Dreieck, etc.)</li> </ul>
<p>NP-vollständige Probleme lösen mit Backtracking</p> <ul style="list-style-type: none"> <li>• Erarbeitung verschiedener NP-vollständiger Probleme</li> <li>• Algorithmische Beschreibung einer Lösungsidee</li> </ul>	<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“ und "Backtracking"(M)</li> <li>• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen(I).</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<ul style="list-style-type: none"> <li>• Rucksackproblem</li> <li>• Irrgartenproblem</li> </ul>
<p>Effiziente Sortierverfahren / Suchverfahren</p> <ul style="list-style-type: none"> <li>• Wiederholung bereits bekannter Sortier- und Suchverfahren als rekursiver Algorithmus</li> <li>• Erarbeitung eines Sortierverfahrens der Laufzeit</li> <li>• <math>O(n \cdot \log(n))</math></li> </ul>	<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 um-</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrationsprogramm zur Visualisierung von Sortierverfahren</li> <li>• Quicksortvisualisierung zur Erarbeitung der Idee</li> <li>• Suchspiel zur Erarbeitung der Binären Suche</li> </ul>

<ul style="list-style-type: none"><li>• Erarbeitung eines Suchverfahrens der Laufzeit <math>O(\log(n))</math>.</li></ul>	<p>gangssprachlich und grafisch dar (D),</p> <ul style="list-style-type: none"><li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und "Backtracking"(M),</li><li>• testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen(I).</li><li>• implementieren und erläutern iterative und</li><li>• rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten)(I),</li><li>• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li><li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li></ul>	
--	--	--

## **Unterrichtsvorhaben Q1-II**

*Thema:* Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen

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

*Zeitbedarf:* 25 Stunden

### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Stapeln am Beispiel dargestellt und die Operationen der Klasse Stack anhand der Abiturklasse erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Dabei werden die Operationen der Datenstruktur Schlange thematisiert und die entsprechende Abiturklasse Queue verwendet.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List gemäß der Abiturklasse eingeführt und in einem Anwendungskontext verwendet. (Je nach thematisierten Anwendungskontexten ist eine andere Reihenfolge bei der Behandlung der oben genannten Datenstrukturen möglich und zulässig.) In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Stack</li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Stack.</li> </ul>	<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>• 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>Visualisierungsprogramm zu dynamischen Datenstrukturen</p>
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Erarbeitung der Funktionalität der Klasse Queue</li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Queue.</li> </ul>	<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>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> </ul>	<p>Visualisierungsprogramm zu dynamischen Datenstrukturen</p>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</li> <li>• Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</li> </ul>	<ul style="list-style-type: none"> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen (D),</li> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> <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>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• testen Programme systematisch anhand von Beispielen .</li> <li>• erläutern Operationen dynamischer (linearer</li> </ul>	<ul style="list-style-type: none"> <li>• Visualisierungsprogramm zu dynamischen Datenstrukturen</li> <li>• Einfache Oberfläche zur Visualisierung der linearen Liste, z. B. die Verwaltung einer Namensliste</li> </ul>

	<ul style="list-style-type: none"><li>• oder nicht-linearer) Datenstrukturen (A),</li></ul>	
Vertiefung / Anwendung einer linearen Datenstruktur im Anwendungskontext.	<ul style="list-style-type: none"><li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li><li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M)</li></ul>	Umsetzung in einem größeren Projekt



## **Unterrichtsvorhaben Q1-III**

*Thema:* Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume

*Leitfragen:* Wie können Daten im Anwendungskontext mit Hilfe binärer Baumstrukturen verwaltet werden? Wie kann dabei der rekursive Aufbau der Baumstruktur genutzt werden? Welche Vor- und Nachteile haben Suchbäume für die geordnete Verwaltung von Daten?

*Zeitbedarf:* 20 Stunden

### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt. Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse BinaryTree der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert. Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum => Inorder-Ausgabe an einem Beispiel verdeutlicht wird.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse Binary-Search-Tree der Vorgaben für das Zentralabitur weitere Klassen oder Methoden in diesem Kontext modelliert und implementiert. Die Suchbäume werden wie zuvor auch grafisch dargestellt.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderer Kontexten weiter geübt.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ul style="list-style-type: none"> <li>• Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</li> <li>• Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</li> </ul>	<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>• 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>Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</li> <li>• Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</li> <li>• Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung</li> <li>• Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</li> </ul>	<ul style="list-style-type: none"> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M),</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> </ul>	<ul style="list-style-type: none"> <li>• Traversierungsverfahren</li> <li>• Projektarbeit Morsebaum</li> </ul>
<p>Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <ul style="list-style-type: none"> <li>• Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</li> <li>• Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm,</li> <li>• grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</li> <li>• Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</li> <li>• Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</li> </ul>	<ul style="list-style-type: none"> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen (D),</li> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> <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</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zum binären Suchbaum</li> <li>• Arbeitsblätter zum Baumsortieren und zu Traversierungsverfahren</li> </ul>

<p>Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>	<p>„Teilen und Herrschen“(M),</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.</li><li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A).</li></ul>	<p>Projektarbeiten zu einem der Themen Termbäume, Morsebäume, Stichwortbaum, Ahnenbaum</p>
--	--	--

## **Unterrichtsvorhaben Q1-IV**

*Thema:* Sicherheit und Datenschutz in Netzstrukturen

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

*Zeitbedarf:* 15 Stunden (Das Thema bietet Vertiefungen darüber hinaus an vielen Stellen an. Ein erste Erprobung des Unterrichtsvorhabens kann über Möglichkeiten hierzu Aufschluss geben und den zeitlichen Bedarf ggf. verändern.)

### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Ausgehend von einer Kommunikation zwischen zwei Kommunikationspartnern über eine einfache Leitung wird die Notwendigkeiten einer Datenübertragung erarbeitet. Die Schichten des TCP/IP-Schichtenmodells werden beispielgebunden erarbeitet (Basisbandübertragungsverfahren, Prüfverfahren, Vermittlungsschicht, Anwendungsprotokoll) und an einer Simulationssoftware getestet. Verschiedene Netzwerk-Topologien werden entwickelt und in Client-Server-Anwendungen simuliert.

Über die Sicherheit von Netzwerkanwendungen wird das Augenmerk auf verschiedene symmetrische und asymmetrische kryptografische Verfahren gelenkt, welche analysiert und erläutert werden. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
Schichten des TCP/IP-Protokolls <ul style="list-style-type: none"> <li>• Erarbeitung der Notwendigkeiten einer Netzwerkkommunikation</li> <li>• Erarbeitung der Schichten des TCP/IP-Protokolls: Ethernet-, Internet-, Transport- und Anwendungsschicht</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A)</li> </ul>	Arbeitsblätter zur Einführung in Netzwerke
Simulation von Netzwerken / Netzwerk-Topologien <ul style="list-style-type: none"> <li>• Erarbeitung der Topologien: Peer-to-Peer, Sterntopologie, Baumtopologie, Vermaschtes Netz</li> <li>• Simulation von Client-Server-Anwendungen</li> <li>• Simulation von Protokollen der Anwendungsschicht (POP3, SMTP, etc.)</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> </ul>	<ul style="list-style-type: none"> <li>• Simulationssoftware FILIUS</li> <li>• Arbeitsblätter und Skript zu FILIUS</li> </ul>
Analyse und Erläuterung kryptografischer Verfahren <ul style="list-style-type: none"> <li>• Erläuterung symmetrischer Verfahren: monoalphabetisch: Cäsar, polyalphabetisch: Vigenère</li> <li>• Erläuterung asymmetrischer Verfahren: RSA, Diffie-Hellman</li> <li>• Analyse der Sicherheit verschiedener Verfahren und Auswirkungen auf den Datenschutz/Urheberrecht</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).</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>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zu kryptografischen Verfahren</li> <li>• CrypTool</li> <li>• Materialien von klicksafe (Zusatzmodule „Nicht alles was geht, ist auch erlaubt“, „Ich bin öffentlich ganz privat“, „Datenschutz im Internet“)</li> </ul>

## **Unterrichtsvorhaben Q2-I**

*Thema:* Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

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

*Zeitbedarf:* 25Stunden

### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Ausgehend von einer konkreten Anwendungssituation entwickeln die Schülerinnen und Schüler Ideen zur Modellierung von Daten und erkennen die Vorzüge von Datenbanksystemen.

In weiteren Anwendungskontexten müssen Datenbanken entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in das Relationale Modell überführt.

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

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

Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. Die Operationen der Relationenalgebra werden mit SQL-Abfragen simuliert.

Anhand von Fallbeispielen werden Probleme bei der Nutzung von Datenbanksystemen aufgezeigt und im Hinblick auf gesellschaftliche Auswirkungen diskutiert. (Die exakte Abfolge der Teilvorhaben sowie die Verzahnung theoretischer und praktischer Anteile kann hierbei je nach Anwendungskontext flexibel angepasst werden.)

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> <li>• Entity-Relationship-Diagramm: Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms. Erläuterung und Modifizierung einer Datenbankmodellierung</li> <li>• Entwicklung einer Datenbank aus einem Datenbankentwurf:</li> <li>• Modellierung eines relationalen Datenbankschemata zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> <li>• Redundanz, Konsistenz und Normalformen: Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation. Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <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>Arbeitsblätter zur Einführung in Datenbanken</p>
<p>Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> <li>• Aufbau von Datenbanken und Grundbegriffe: Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li> <li>• SQL-Abfragen: Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle</li> <li>• Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• modifizieren eine Datenbankmodellierung (M),</li> <li>• bestimmen Primär- und Sekundärschlüssel (M)</li> <li>• analysieren und erläutern eine Datenbankmodellierung (A)</li> <li>• ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D)</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>• erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nut-</li> </ul>	

<p>Fragestellungen (JOIN, 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)</p> <ul style="list-style-type: none"> <li>• Vertiefung an einem weiteren Datenbankbeispiel: Vertiefungen am Beispiel der Relationenalgebra</li> </ul>	<p>zung (A)</p>	
<p>Gesellschaftliche Auswirkungen der Nutzung von Datenbanksystemen</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> <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>	<ul style="list-style-type: none"> <li>• Fallbeispiele zur Nutzung von Datenbanksystemen</li> <li>• Spiel zum Missbrauch von Daten: DataDealer</li> <li>• (<a href="http://datadealer.com/de">http://datadealer.com/de</a>)</li> </ul>



## **Unterrichtsvorhaben Q2-II**

*Thema:* Endliche Automaten und Formale Sprachen

*Leitfragen:* Wie kann man endliche Automaten genau beschreiben? Wie können endliche Automaten modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, Automaten und Grammatiken?

*Zeitbedarf:* 25 Stunden

### *Absprachen zur vorhaben-bezogene Konkretisierung:*

Ausgehend von einem konkreten Anwendungsbeispiel entwickeln die Schülerinnen und Schüler das Modell der Grammatik einer formalen Sprache und das Modell des endlichen Automaten. Die Schülerinnen und Schüler überführen Automaten in verschiedene Darstellungsformen und ermitteln die akzeptierte Sprache eines Automaten (z. B. in Form von regulären Ausdrücken). An einem Beispiel wird ein nichtdeterministischer Akzeptor als Alternative gegenüber einem entsprechenden deterministischen Akzeptor eingeführt.

Der Zusammenhang zwischen endlichen Automaten und regulären Grammatiken wird durch die Entwicklung allgemeingültiger Verfahren zur Transformation zwischen Automat und Grammatik dargestellt. Die Unzulänglichkeit endlicher Automaten und regulärer Grammatiken wird an Beispielen verdeutlicht.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Einführung in Automaten/Grammatiken</p> <ul style="list-style-type: none"> <li>• Grammatiken: Grammatik einer natürlichen Sprache Grammatik einer künstlichen Sprache Idee des Parsens</li> <li>• Automaten: erkennender Automat zu Symbolen einer Sprache Modell des endlichen Automaten Darstellungsformen</li> <li>• Sprache eines Automaten als regulärer Ausdruck nichtdeterministische Automaten</li> </ul>	<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</li> <li>• stellen endliche Automaten in Tabellen oder 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> </ul>	<p>Taschenrechner</p>
<p>Zusammenhang zwischen endlichen Automaten und regulären Grammatiken</p> <ul style="list-style-type: none"> <li>• reguläre Grammatik: Definition Anwendungen</li> <li>• Zusammenhang zu endlichen Automaten</li> <li>• Grenzen der endlichen Automaten/regulären Grammatiken</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <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> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> <li>• entwickeln zur akzeptierten Sprache eines Automaten</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D),</li> <li>• zeigen Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A)</li> </ul>	<p>Arbeitsblätter</p>

## ***Unterrichtsvorhaben Q2-III***

*Thema:* Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

*Leitfragen:* 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?

*Zeitbedarf:* 10 Stunden

### ***Absprachen zur vorhaben-bezogene Konkretisierung:***

Anhand einer von-Neumann-Architektur und einem maschinennahen Programm wird die prinzipielle Arbeitsweise von Computern verdeutlicht. Ausgehend von den prinzipiellen Grenzen endlicher Automaten liegt die Frage nach den Grenzen von Computern bzw. nach Grenzen der Automatisierbarkeit nahe. Mit Hilfe einer entsprechenden Java-Methode wird plausibel, dass es unmöglich ist, ein Informatiksystem zu entwickeln, das für jedes beliebige Computerprogramm und jede beliebige Eingabe entscheidet ob das Programm mit der Eingabe terminiert oder nicht (Halteproblem). Anschließend werden Vor- und Nachteile der Grenzen der Automatisierbarkeit angesprochen und der Einsatz von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen beurteilt.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>Von-Neumann-Architektur und die Ausführung maschinennaher Programme:</p> <ul style="list-style-type: none"> <li>• prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</li> <li>• einige maschinennahe Befehlen und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</li> <li>• Analyse und Erläuterung der Funktionsweise eines</li> <li>• einfachen maschinennahen Programms</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-NeumannArchitektur“ (A),</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A)</li> </ul>	<ul style="list-style-type: none"> <li>• geeigneter Modellrechner</li> <li>• WinAli</li> <li>• Microprocessor Simulator</li> <li>• Computerclub „Papier-CPU“</li> </ul>
<p>Grenzen der Automatisierbarkeit:</p> <ul style="list-style-type: none"> <li>• Vorstellung des Halteproblems</li> <li>• Unlösbarkeit des Halteproblems</li> <li>• Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</li> </ul>		

## **Kursarbeiten bzw. Klausuren**

Kursarbeiten bzw. Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse einer vorausgegangenen Unterrichtsreihe. Sie sind so anzulegen, dass Sachkenntnisse und methodische Fertigkeiten nachgewiesen werden können. Sie bedürfen einer angemessenen Vorbereitung und verlangen klare Aufgabenstellungen. Im Umfang und Anforderungsniveau sind Kursarbeiten bzw. Klausuren abhängig von den kontinuierlich ansteigenden

Anforderungen entsprechend dem Lehrplan.

Es ist darauf zu achten, dass nicht nur die Richtigkeit der Ergebnisse und die inhaltliche Qualität, sondern auch die angemessene Form der Darstellung unabdingbare Kriterien der Bewertung der geforderten Leistung sind. Am Pestalozzi-Gymnasium werden die Kursarbeiten bzw. Klausuren in der Regel nach einem vorab festgelegten Punkteschema bewertet. Dabei ist eine glatt ausreichende Leistung bei 45% der Punktzahl erreicht worden. Die übrigen Notenstufen ergeben sich dann dadurch, dass für jede Notenstufe Intervalle der erreichten Punkte gebildet werden, die in der Regel gleich groß sind:

ab % Punkte

ab %	Punkte
0,00%	0
<b>20,00%</b>	<b>1</b>
<b>26,67%</b>	<b>2</b>
<b>33,33%</b>	<b>3</b>
40,00%	4
45,00%	5
50,00%	6
55,00%	7
<b>60,00%</b>	<b>8</b>
<b>65,00%</b>	<b>9</b>
70,00%	10
75,00%	11
80,00%	12
<b>85,00%</b>	<b>13</b>
<b>90,00%</b>	<b>14</b>
<b>95,00%</b>	<b>15</b>

## **Mitarbeit im Unterricht**

Der Beurteilungsbereich „Mitarbeit im Unterricht“ erfasst die Qualität und Kontinuität der Beiträge, die die Schülerinnen und Schüler im Unterricht erbringen. Diese Beiträge sollen unterschiedliche mündliche und schriftliche Formen in enger Bindung an die Aufgabenstellung, die inhaltliche Reichweite und das Anspruchsniveau der jeweiligen Unterrichtseinheit umfassen.

Bei den mündlichen Leistungen im Unterricht sind zu bewerten:

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Mitarbeit in Partner- und Gruppenarbeitsphase
- Führen eines Lerntagebuchs

Neben der Richtigkeit, Vollständigkeit und Komplexität der Gedankengänge sind die der Altersstufe angemessene sprachliche Darstellung und die Verwendung der Fachsprache von Bedeutung.

Bei der Unterrichtsgestaltung sind den Schülerinnen und Schülern hinreichend Möglichkeiten zur Mitarbeit zu eröffnen, z.B. durch

- praktische Leistungen am Computer als Werkzeug im Unterricht,
- Protokolle und Referate,
- Führen eines Lerntagebuchs,
- Projektarbeit (oft in Form von Gruppenarbeit),
- Lernerfolgsüberprüfungen und schriftliche Übungen.

### ***Bildung der Zeugnisnote***

In die Note gehen alle im Unterricht erbrachten Leistungen ein. Dabei nehmen die Beurteilung der Kursarbeiten bzw. Klausuren den gleichen Stellenwert wie die Leistungen im Bereich der Mitarbeit im Unterricht ein. Zudem ist bei der Notenfindung die individuelle Lernentwicklung der Schülerinnen und Schüler angemessen zu berücksichtigen.