



HILDEGARDIS  
SCHULE  
HAGEN

# Lehrplan und Kriterien zur Leistungsbewertung

für das Fach

# Informatik

## Inhaltsverzeichnis

<b>1. RAHMENBEDINGUNGEN DER FACHLICHEN ARBEIT .....</b>	<b>3</b>
<b>2. ENTSCHEIDUNGEN ZUM UNTERRICHT .....</b>	<b>5</b>
2.1 UNTERRICHTSVORHABEN .....	5
2.2 ÜBERSICHT ÜBER DIE ZUR ERZIELENDEN KOMPETENZEN .....	6
2.3 FORMEN DER KOMPETENZÜBERPRÜFUNG .....	7
2.4 ÜBERSICHT ÜBER DIE INHALTSFELDER .....	8
2.5 KONKRETISIERTE UNTERRICHTSVORHABEN .....	9
<b>2.5.1 Einführungsphase .....</b>	<b>9</b>
<b>2.5.2 Qualifikationsphase .....</b>	<b>20</b>
2.6 GRUNDSÄTZE DER FACHMETHODISCHEN UND FACHDIDAKTISCHEN ARBEIT .....	38
2.7 GRUNDSÄTZE DER LEISTUNGSBEWERTUNG UND LEISTUNGSRÜCKMELDUNG .....	39
2.8 LEHR- UND LERNMITTEL .....	41
<b>3. ENTSCHEIDUNGEN ZU FACH- UND UNTERRICHTSÜBERGREIFENDEN FRAGEN .....</b>	<b>41</b>
<b>4. QUALITÄTSSICHERUNG UND EVALUATION .....</b>	<b>42</b>

## 1. Rahmenbedingungen der fachlichen Arbeit

Die Hildegardis-Schule Hagen als katholisches Gymnasium in Trägerschaft des Erzbistums Paderborn hat traditionell einen großen Einzugsbereich. Ihre Schülerschaft kommt zwar zum überwiegenden Teil aus Hagen, aber nicht unerheblich von umliegenden Städten und Gemeinden wie z.B. Breckerfeld, Schalksmühle, Ennepetal, Wetter, Witten, Schwerte. Entsprechend heterogen sind das Vorwissen und der Kenntnisstand bezogen auf die Informatik. Aber auch der private Zugang oder Bezug zu Computern oder entsprechenden Medien gestaltet sich aufgrund des inzwischen heterogener gewordenen sozialen Umfelds und der damit oft verbundenen materiellen Grundlagen sehr unterschiedlich. Die Schule bietet über den Schulträger nahezu kostenlose Lizenzen für das aktuelle Office-Paket der Firma Microsoft an. Daneben hat die Fachschaft Informatik beschlossen, ausschließlich Software im Unterricht einzusetzen, die auch als Freeware erhältlich ist, um den Schülerinnen und Schülern auch das kostenlose Arbeiten an einem heimischen PC zu ermöglichen. Mit dem Schulverein wird in Kürze geklärt werden, ob durch ihn für Lernende aus finanziell beschränkten Haushalten entsprechende Rechner leihweise kostenlos zur Verfügung gestellt werden können.

Das Fach Informatik wird zunächst als selbstständiges Fach nur im Rahmen der gymnasialen Oberstufe als Grundkurs angeboten. In der Sekundarstufe I ist es in ein neues Konzept für den Wahlpflichtbereich II mit dem Namen „IFO – Forschen mit Informatik“ integriert. Dabei handelt es sich um ein fächerübergreifendes und fächerverbindendes Angebot der naturwissenschaftlichen Fächer Biologie, Chemie und Physik, bei dem Teilaspekte der Informatik bei der Erfassung, Darstellung und Auswertung von Messungen oder der Simulation von Vorgängen zum Tragen kommen. Es ist angedacht, den Informatik-Anteil durch ein neues Angebot in Robotik unter Einsatz von LEGO™-Educator dort zu erhöhen. Wegen des zurzeit noch stark experimentellen Charakters und des nur sehr geringen Anteils an Informatik wird aktuell hier kein Lehrplan für Informatik in der Sek I gesondert aufgeführt.

Für den Unterricht steht zurzeit ein Raum mit 30 Arbeitsplätzen in Gruppen zu je 6 Einheiten zur Verfügung. Unter der pädagogischen Oberfläche von MTS™-Reinhard können dort die Schülerinnen und Schüler mit den verschiedenen Programmen einzeln oder in Gruppen arbeiten. Die Oberfläche ermöglicht es dabei, dass sowohl Demonstrationen des Lehrenden als auch Präsentationen einzelner Schüler oder Schülergruppen auf allen Rechnern dargestellt und verfolgt werden können.

Die in den letzten Jahren zunehmend verschärften Wahlbedingungen für die gymnasiale Oberstufe haben die Einrichtung von stabilen Kursen im Fach Informatik erschwert. Als Nicht-Naturwissenschaft besitzt es keinen verpflichtenden Charakter und steht zudem in immer stärker werdender Konkurrenz zum Fach Mathematik, welches aufgrund der Abiturbedingungen zunehmend als Abiturfach das Aufgabenfeld III abdeckend gewählt wird. So konnten in den Schuljahren 2014/15 und 2015/16 nach der Einführungsphase wegen zu geringer Wahlzahlen keine Kurse in der Qualifikationsphase eingerichtet werden. Es bleibt abzuwarten, ob ein verstärkter praxisorientierter Unterricht und ein noch intensiveres Eingehen auf die Fähigkeiten, Interessen und Wünsche der Schülerinnen und Schüler

hier zu einer Stabilisierung im Sinne des Zustandekommens eines durchgängig bis zum Abitur führenden Kurses führen wird.

Gerade das Fach Informatik kann in Zeiten einer von Medien und Datenflut geprägten Umwelt einen wichtigen Erziehungsbeitrag für die Entwicklung der jungen Menschen leisten. Indem die Informatik hinter die Fassade blickt, die Mechanismen von Datensammlung, Datenspeicherung und Datenverarbeitung aufzeigt, auf die Problematik von Datensicherheit und Eigenverantwortung aufmerksam macht und sowohl die Fähigkeiten moderner Rechner als auch ihre prinzipiellen Grenzen aufzeigt, kann sie wichtige Informationen liefern und Impulse setzen für einen kritischen, selbstverantworteten und gesellschaftspolitisch engagierten und ethisch vertretbaren Umgang mit Medien und Daten. Die Anbindung an das Leitbild und die Erziehungsziele der Schule, wie sie im Schulprogramm formuliert sind, ist unmittelbar gegeben. Gerade in der Qualifikationsphase bieten sich zahlreiche fächerverbindende und fächerübergreifende Möglichkeiten zur Zusammenarbeit mit Fächern wie Deutsch, Politik/Sozialwissenschaften, Erziehungswissenschaften, aber auch den Naturwissenschaften oder Religion an. Sie sollten im Rahmen von Schulentwicklung konkretisiert und nach Möglichkeit in den jeweiligen Kernlehrplänen verankert werden, was sicher auch zur Stabilisierung des Faches im Kanon der Fächer in der gymnasialen Oberstufe beiträge und der eigentlichen Bedeutung des Faches gerecht würde.

### **Funktionsinhaber/innen der Fachgruppe**

Fachkonferenzvorsitzender:

Klaus Nelius

Kontaktlehrer/in für die Zusammenarbeit mit außerschulischen Partnern (z.B. Tu Dortmund):

Sabine Schiller, Markus Plöger

Ansprechpartner bei technischen Fragen:

Markus Plöger, Klaus Nelius

## 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. Diese werden zur Verdeutlichung im Kapitel 2.1.1 noch einmal in gebündelter Form aufgelistet.

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 gemäß Fachkonferenzbeschluss 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. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Berufsvorbereitungswoche, Berufspraktikum, Fahrten im Rahmen des Austauschprogramms, andere Formen der Exkursionen, Abiturprüfungszeiten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans nur ca. 75 Prozent der Bruttounterrichtszeit verplant.

Die Fachkonferenzbeschlüsse zu den „konkretisierten Unterrichtsvorhaben“ dienen zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln. Sie sind deshalb für alle Mitglieder der Fachkonferenz Bindekraft bindend, enthalten aber gleichzeitig auch notwendige Freiräume durch Beispiele für mögliche Projekte, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie potentiellen neuen Kolleginnen und Kollegen dienen die Übersichten vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind.

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

## 2.2 Übersicht über die zur erzielenden Kompetenzen

Die in den Richtlinien aufgeführten Kompetenzen sind nachstehend aufgelistet und werden den jeweiligen konkreten Ausführungen den zu entwickelnden Kompetenzen zugeordnet:

- **Argumentieren (A)**

*Die Schülerinnen und Schüler*

- *erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme.*
- *analysieren und erläutern informatische Modelle.*
- *analysieren und erläutern Computerprogramme.*
- *beurteilen die Angemessenheit informatischer Modelle.*

- **Modellieren (M)**

*Die Schülerinnen und Schüler*

- *konstruieren zu kontextbezogenen Problemstellungen informatische Systeme.*
- *modifizieren und erweitern informatische Systeme.*

- **Implementieren (I)**

*Die Schülerinnen und Schüler*

- *implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme.*
- *modifizieren und erweitern Computerprogramme.*
- *testen und korrigieren Computerprogramme.*

- **Darstellen und Interpretieren (D)**

*Die Schülerinnen und Schüler*

- *interpretieren Daten und erläutern Beziehungen und Abläufe, die in Form von textuellen und graphischen Darstellungen gegeben sind.*
- *überführen gegebenen textuelle und graphische Darstellungen informatorischer Zusammenhänge in die jeweils andere Darstellungsform.*
- *stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Graphiken dar.*

- **Kommunizieren und Kooperieren (K)**

*Die Schülerinnen und Schüler*

- *verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),*
- *präsentieren Arbeitsabläufe und -ergebnisse (K),*
- *kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),*
- *nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung*

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

### 2.3 Formen der Kompetenzüberprüfung

Die Kompetenzerwartungen des Lehrplans ermöglichen eine Vielzahl von Überprüfungsformen. Diese sind nachstehend aufgeführt und werden von der Lehrkraft sequenzbezogen und individuell bei den einzelnen Unterrichtsvorhaben variabel eingesetzt. Sie werden deshalb nicht explizit neben den zu erreichenden Kompetenzen aufgeführt. Daneben können noch bis zu 2 kurze schriftliche Übungen je Halbjahr geschrieben werden, in denen das Erreichen der beschriebenen Kompetenzen für die gesamte Lerngruppe überprüft wird.

Folgende Überprüfungsformen sehen die Kernlehrpläne für das Fach Informatik vor:

- |  |              |
|--|--------------|
| <i>Überprüfungsform</i>  | <i>I:</i>    |
| Analyse und Eingrenzung einer kontextbezogenen Problemstellung und Entwicklung eines Modells oder Teilmodells mit erläuternden Begründungen der Entwurfsentscheidung |              |
| <i>Überprüfungsform</i>  | <i>II:</i>   |
| Analyse, Erläuterung und Modifikation eines vorgegebenen informatischen Modells sowie die vergleichende Beurteilung unterschiedlicher Entwürfe                       |              |
| <i>Überprüfungsform</i>  | <i>III:</i>  |
| Vollständige oder teilweise Implementation einer bereits modellierten Problemstellung  |              |
| <i>Überprüfungsform</i>  | <i>IV:</i>   |
| Entwurf und formale Darstellung von Algorithmen zu einer vorgegebenen informatischen Problemstellung   |              |
| <i>Überprüfungsform</i>  | <i>V:</i>    |
| Analyse und Erläuterung von vorgegebenen Algorithmen oder Programmausschnitten   |              |
| <i>Überprüfungsform</i>  | <i>VI:</i>   |
| Interpretation gegebener textueller, graphischer oder formaler Darstellungen informatischer Zusammenhänge und deren Überführung in eine andere Darstellungsform      |              |
| <i>Überprüfungsform</i>  | <i>VII:</i>  |
| Darstellung, Erläuterung und sachgerechte Anwendung von informatischen Begriffen, Verfahren und Lösungsstrategien  |              |
| <i>Überprüfungsform</i>  | <i>VIII:</i> |
| Analyse und Beurteilung einer Problemlösung oder eines Informatiksystems nach vorgegebenen oder eigenen Kriterien  |              |
| <i>Überprüfungsform</i>  | <i>IX:</i>   |
| Analyse und Bewertung des Einsatzes eines Informatiksystems in Bezug auf ethische, rechtliche oder gesellschaftliche Fragestellungen                                 |              |

## 2.4 Übersicht über die Inhaltsfelder

Folgende Inhaltsfelder sind für die Einführungsphase obligatorisch und werden zur Vereinfachung jeweils nur als Verweis bei den Unterrichtsvorhaben aufgeführt:

1. Daten und ihre Strukturierung (DS)
2. Algorithmen (AL)
3. Formale Sprachen und Automaten (FSA)
4. Informatiksysteme (IS)
5. Informatik, Mensch und Gesellschaft (IMG)



## 2.5 Konkretisierte Unterrichtsvorhaben

## 2.5.1 Einführungsphase

<b>Unterrichtsvorhaben I</b>		
<b>Thema:</b> Nutzung von Informatiksystemen / grundlegende Begrifflichkeiten		
<b>Inhaltsfelder:</b> IS, IMG		
<b>Zeitbedarf:</b> 6 Unterrichtsstunden		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<p><b>1. Information, deren Kodierung und Speicherung</b></p> <p>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</p> <p>(b) Darstellung von Informationen in Schrift, Bild und Ton</p> <p>(c) ©Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</p> <p>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</p> <p><b>2. Informations- und Datenübermittlung in Netzen</b></p> <p>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</p> <p>(b) Informatische Kommunikation in Rechnernetzen am Beispiel <i>des Schulnetzwerks</i> (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</p> <p>(c) ©Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</p> <p>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),</li> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p><i>Sprache:</i> <i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Wingdings)</p> <p><i>Kunst:</i> <i>Beispiel:</i> Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken</p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>



<b>Unterrichtsvorhaben II</b>		
<b>Thema:</b> Grundlagen objektorientierter Analyse, Modellierung und Implementierung		
<b>Inhaltsfelder:</b> DS, AL		
<b>Zeitbedarf:</b> 9 Unterrichtsstunden		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<p><b>1. Identifikation von Objekten</b></p> <p>(a) Am Beispiel eines lebenswelt-nahen Beispiels werden Objekte im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</p> <p>(c) ©Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p> <p><b>2. Analyse von Klassen didaktischer Lernumgebungen</b></p> <p>(a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</p> <p>(b) Teilanalyse der Klassen der verwendeten didaktischen Lernumgebung</p> <p><b>3. Implementierung statischer Szenen mit Graphikobjekten</b></p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Konzeption einer Szene mit Graphik-Objekten</p> <p>(c) ©Deklaration und Initialisierung von Objekten</p> <p>(d) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B.</p>	<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>stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>stellen den Zustand eines Objekts dar (D).</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p><i>Biologie</i> <i>Beispiel: Vogelschwarm</i></p> <p>Schülerinnen und Schüler betrachten einen Vogelschwarm als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.</p> <p><i>Mathematik</i> <i>Beispiel: regelmäßige Flächen, Körper</i></p> <p>Schülerinnen und Schüler betrachten bekannte Flächen oder Körper als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.</p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>

Farbe, Position, Drehung)		

<b>Unterrichtsvorhaben III</b>		
<b>Thema:</b> Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen		
<b>Inhaltsfelder:</b> DS, AL		
<b>Zeitbedarf:</b> 18 Unterrichtsstunden		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<p><b>1. Bewegungsanimationen am Beispiel einfacher grafischer Objekte (GO)</b></p> <p>(a) Kontinuierliche Verschiebung eines GO mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationschleife</p> <p>(c) ©Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(d) Berechnung von Abständen zwischen GO mit Hilfsvariablen</p> <p>(e) Meldungen zur Kollision zweier GO mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)</p> <p><b>2. Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte (GO)</b></p> <p>(a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife)</p> <p>(b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays)</p> <p>(c) ©Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden</p> <p>(d) Vertiefung: Verschiedene Feldbeispiele</p> <p><b>3. Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte</b></p> <p>(a) Modellierung eines Simulationsprogramms mit eigenen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und 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>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p>algorithmisches Vorgehen in den Naturwissenschaften</p> <p><b>außerschulische Lernorte:</b></p> <p>keine</p> <p><b>Formen der Kompetenzüberprüfung:</b></p> <p>s.o.</p>

<p>Klassen, die sich selbst mit Hilfe von einfachen GO zeigen mit Hilfe eines Implementationsdiagramms</p> <p>(b) Implementierung eigener Methoden mit und ohne Parameterübergabe</p> <p>(c) Realisierung von Zustandsvariablen</p> <p>(d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten</p> <p>(e) Animation mit Hilfe des Aufrufs von selbstimplementierten Methoden</p> <p>(f) Vertiefung: Weitere Projekte</p>	<ul style="list-style-type: none"> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	
---	---	--

## Unterrichtsvorhaben IV

**Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen**

**Inhaltsfelder: DS, AL**

**Zeitbedarf: 18 Unterrichtsstunden**

<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<p><b>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</b></p> <p>(a) Einführung der Elemente einer ereignisgesteuerten Oberfläche (Schalter, Taster, Textfelder usw.)</p> <p>(b) Einführung von Mausgesteuerten Ereignisanwendungen, ereignisorientierte Programmierung</p> <p>(c) ©Steuerung einfacher grafischer Objekte durch Ereignisbearbeiter.</p> <p><b>2. Entwicklung eines Spiels mit der Notwendigkeit von Kollisionkontrollen zwischen zwei oder mehr grafischen Objekten</b></p> <p>(a) Modellierung des Spiels ohne Berücksichtigung der Kollision mit Hilfe eines Implementationsdiagramms</p> <p>(b) Dokumentation der Klassen des Projekts</p> <p>(c) ©Implementierung eines Prototypen ohne Kollision</p> <p>(d) Ergänzung einer Kollisionsabfrage durch zusätzliche Assoziationsbeziehungen in Diagramm, Dokumentation und Quellcode</p> <p>(e) Verallgemeinerung der neuen Verwendung von Objektreferenzen</p> <p>(f) Vertiefung: Entwicklung weiterer Spiele und Simulationen mit vergleichbarer Grundmodellierung</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</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p>algorithmisches Vorgehen in den Naturwissenschaften</p> <p><b>außerschulische Lernorte:</b></p> <p>keine</p> <p><b>Formen der Kompetenzüberprüfung:</b></p> <p>s.o.</p> <p>Beispiel: Ufospiel Die Schülerinnen und Schüler entwickeln die Simulation eines Ufos, das Asteroiden ausweichen soll, mit denen eine Kollision möglich ist.</p> <p>Beispiel: Billardkugeln Die Schülerinnen und Schüler entwickeln ein Spiel, bei dem eine Billardkugel gegen andere gespielt werden kann</p> <p>Beispiel: Autospiel Die Schülerinnen und Schüler entwickeln ein Spiel, bei dem ein Auto mausgesteuert durch einen Hindernisparcour gelenkt werden kann.</p>

<p><b>3. Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden)</b></p> <p>(a) Analyse und Erläuterung einer Basisversion der grafischen Klasse</p> <p>(b) Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode)</p> <p>(c) ©Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</p> <p><b>4. Entwicklung einer komplexeren Simulation mit grafischen Elementen, die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden)</b></p> <p>(a) Analyse und Erläuterung einer einfachen grafischen Animationsklasse</p> <p>(b) Spezialisierung der Klasse zu Unterklassen mit verschiedenen Animationen durch Überschreiben der entsprechenden Animationsmethode</p> <p>(c) ©Reflexion des Prinzips der späten Bindung</p> <p>(d) Vertiefung: Entwicklung eines vergleichbaren Projekts mit einer (abstrakten) Oberklasse</p>	<p>von Vererbung (M),</p> <ul style="list-style-type: none"> <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>• modifizieren einfache Algorithmen und Programme (I),</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>Beispiel: Schneemann</p> <p>Die Schülerinnen und Schüler erstellen eine Simulation von Schneemännern, die unterschiedliche Kopfbedeckungen tragen.</p> <p><i>Beispiel: Flummibälle</i> Die Schülerinnen und Schüler entwickeln eine Simulation von Flummibällen, bei der unterschiedliche Bälle unterschiedliche Bewegungen durchführen.</p> <p><i>Beispiel: Weihnachtsbaum</i> Die Schülerinnen und Schüler entwickeln eine Simulation eines Weihnachtsbaums mit Hilfe einer abstrakten Klasse Schmuck.</p>
--	---	---



<b>Unterrichtsvorhaben V</b>		
<b>Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</b>		
<b>Inhaltsfelder: DS, AL, IMG</b>		
<b>Zeitbedarf: 12 Unterrichtsstunden</b>		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<p><b>1. Das Sortierproblem und die Erarbeitung eines elementaren Sortierverfahrens</b></p> <p>(a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) ©Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p> <p><b>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) ©Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs Analyse des weiteren Sortier-</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),</li> <li>• entwerfen einen weiteren Algorithmus zum Sortieren (M),</li> <li>• analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p> <p><i>Beispiele:</i></p> <p>Sortieren durch Auswählen,</p> <p>Sortieren durch Vertauschen, Quicksort Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p>

<p>algorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)</p> <p><b>3. Binäre Suche auf sortierten Daten</b></p> <p>(a) Suchaufgaben im Alltag und im Kontext informatischer Systeme</p> <p>(b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche</p> <p>(c) ©Effizienzbetrachtungen zur binären Suche</p>		<p><i>Beispiel:</i></p> <p>Simulationsspiel zur binären Suche nach Tischtennisbällen</p> <p>Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p>
---	--	--

<b>Unterrichtsvorhaben VI</b>		
<b>Thema: Geschichte der digitalen Datenverarbeitung / Grundlagen des Datenschutzes</b>		
<b>Inhaltsfelder: IMG</b>		
<b>Zeitbedarf: 12 Unterrichtsstunden</b>		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<p><b>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</b></p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none"> <li>• „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“</li> <li>• „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“</li> <li>• „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“</li> <li>• „Kodieren von Texten und Bildern: ASCII, RGB und mehr“</li> <li>• „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“</li> </ul> <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p> <p><b>2. Vertiefung des Themas Datenschutz</b></p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(c) ©Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),</li> <li>• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),</li> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p>Gesellschaftswissenschaft, Politik</p> <p>Geschichte, Deutsch</p> <p><b>außerschulische Lernorte:</b></p> <p>evtl. Besuch des Nixdorf-Museums in Paderborn</p> <p><b>Formen der Kompetenzüberprüfung:</b></p> <p>s.o.</p>

## 2.5.2 Qualifikationsphase

<b>Unterrichtsvorhaben I (Q1)</b>		
<b>Thema:</b> Wiederholung der objektorientierten Modellierung und Programmierung <b>Leitfrage:</b> <i>Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen?</i> <i>Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen?</i> <b>Inhaltsfelder:</b> DS, AL <b>Zeitbedarf:</b> 8 Unterrichtsstunden		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
<b>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</b> (a) Analyse der Problemstellung (b) Analyse der Modellierung (Implementationsdiagramm) (c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) (d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) (e) Dokumentation von Klassen (e) Implementierung der Anwendung oder von Teilen der Anwendung Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung)	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (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>	<b>fachübergreifende Bezüge:</b>  <b>Beispiel: Parkplatz</b> Ein Parkplatz ist belegt von vielen verschiedenen Fahrzeugen, die sich in der Marke, der Farbe, der Größe und weiteren Attributen unterscheiden. Auch Kleinlaste oder LKW können darunter sein. Alle Fahrzeuge haben gemeinsame Merkmale und können vor- und zurück bewegt werden. Daher liegt es nahe, dass entsprechende Attribute und Methoden in mindestens einer Oberklasse realisiert werden.  <b>außerschulische Lernor-</b>

	<ul style="list-style-type: none"> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (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><b>te:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>
--	---	---

## Unterrichtsvorhaben II (Q1)

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

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

**Inhaltsfelder:** DS, AL

**Zeitbedarf:** 8 Unterrichtsstunden

Inhaltsbezogene Kompetenzen	Prozessbezogenen Kompetenzen (Schwerpunkte)	Vereinbarungen der Fachkonferenz
<p>Auf der Basis einer Klasse Node mit dem Attribut item der allgemeinen Klasse Object und zwei Zeigern vom Datentyp Node wird zunächst der Stapel als doppelt verkettete Datenstruktur mit dem Attribut head und den Methoden eines Stapels (push, top, pop, isEmpty) eingeführt. Die Klassen Schlange und Liste ergeben sich durch Vererbung und geeignete Erweiterungen durch weitere Zeigerattribute wie tail und actual und den typischen Methoden der Klassen, die in den Abiturvorgaben detailliert aufgeführt sind.</p>		
<p><b>1. Die Datenstruktur Stack (Stapel) im Anwendungskontext unter Nutzung der Klasse Node (Knoten)</b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) ©Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p> <p><b>2. Die Datenstruktur Queue (Schlange) im Anwendungskontext unter Nutzung der Vererbung der Klasse Stack</b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) ©Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p> <p><b>3. Die Datenstruktur lineare</b></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modifizieren Algorithmen und Pro-</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p><b>Beispiel: Heftstapel</b></p> <p>In einem Heftstapel soll das Heft einer Schülerin gefunden werden.</p> <p><b>Beispiel: Patientenwarteschlange</b> Sobald ein Patient in einer Praxis eintrifft, werden Name und Krankenkasse erfasst. Die Verwaltung der Warteschlange geschieht über eine Klasse Wartezimmer. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird.</p> <p><b>Beispiel: Abfahrtslauf</b></p> <p>Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste einge-</p>

<p><b>Liste im Anwendungskontext unter Nutzung der Klasse List</b></p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p> <p><b>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</b></p>	<p>gramme (I),</p> <ul style="list-style-type: none"> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (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> <li>• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).</li> </ul>	<p>ordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Teilnehmer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p> <p><b>Beispiel: Terme in Postfix-Notation</b></p> <p>Die sog. UPN (<i>Umgekehrt-Polnische-Notation</i>) bzw. <i>Postfix-Notation</i> eines Terms setzt den Operator hinter die Operanden. Um einen Term aus der gewohnten Infixschreibweise in einen Term in UPN umzuwandeln oder um den Wert des Terms zu berechnen, kann ein Stack verwendet werden.</p> <p><b>Beispiel: Rangierbahnhof</b></p> <p>Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzturnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufstei-</p>
---	--	---

		<p>gend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann.</p> <p><b>Weitere Beispiele:</b> vgl. Abituraufgaben mit Bezug zu diesem Thema</p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>
--	--	--

### Unterrichtsvorhaben III (Q1)

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

**Leitfrage:** *Wie kann man gespeicherte Informationen günstig (wieder-) finden?*

**Inhaltsfelder:** DS, AL

**Zeitbedarf:** 12 Unterrichtsstunden

**Inhaltsbezogene Kompetenzen**

**Prozessbezogenen Kompetenzen (Schwerpunkte)**

**Vereinbarungen der Fachkonferenz**

In einem Anwendungskontext werden zunächst Informationen in einer linearen Liste bzw. einem Feld gesucht. Hierzu werden Verfahren entwickelt und implementiert bzw. analysiert und erläutert, wobei neben einem iterativen auch ein rekursives Verfahren thematisiert wird und mindestens ein Verfahren selbst entwickelt und implementiert wird. Die verschiedenen Verfahren werden hinsichtlich Speicherbedarf und Zahl der Vergleichsoperationen miteinander verglichen.

Anschließend werden Sortierverfahren entwickelt und möglichst implementiert (ebenfalls für lineare Listen und Felder). Hierbei soll auch ein rekursives Sortierverfahren, z.B. Quicksort, kennengelernt werden. Eine mögliche Implementation von Quicksort sowie das Sortieren durch Einfügen werden analysiert und erläutert. Falls diese Verfahren vorher schon entdeckt wurden, sollen sie hier wiedererkannt werden. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.



<p><b>1. Suchen von Daten in Listen und Arrays</b></p> <p>(a) Lineare Suche in Listen und in Arrays</p> <p>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</p> <p>(c) ©Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p> <p><b>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</b></p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) ©Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</p> <p><b>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</b></p> <p>(a) Grafische Veranschaulichung der Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) ©Beurteilung der Effizienz der beiden Sortierverfahren</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A)</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>• nutzen Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> </ul>	<p><b>fachübergreifende Bezüge:</b></p> <p>algorithmisches Vorgehen in den Naturwissenschaften</p> <p><b>Beispiel: Karteiverwaltung</b></p> <p>Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p> <p><b>Beispiel: Bundesjugendspiele</b></p> <p>Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken.</p> <p>Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin heraussuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p> <p><b>außerschulische Lernorte:</b></p> <p>keine</p> <p><b>Formen der Kompetenzüberprüfung:</b></p> <p>s.o.</p>
--	---	---

	<ul style="list-style-type: none"><li>• testen Programme systematisch anhand von Beispielen (I),</li><li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li></ul>	
--	--	--

## Unterrichtsvorhaben IV (Q1)

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

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

**Inhaltsfelder:** DS, IS, IMG

**Zeitbedarf:** 21 Unterrichtsstunden

**Inhaltsbezogene Kompetenzen**

**Prozessbezogenen Kompetenzen (Schwerpunkte)**

**Vereinbarungen der Fachkonferenz**

In einem Anwendungskontext soll eine Datenbank 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 der Anwendungssituation Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Weitere Entity-Relationship-Modelle werden erstellt, interpretiert und erläutert, modifiziert und in Datenbankschemata überführt.

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

An einem Beispiel wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert. Ausgehend von einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

**1. Modellierung von relationalen Datenbanken**  
 (a) Entity-Relationship-Diagramm

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

Die Schülerinnen und Schüler

- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),
- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),

**fachübergreifende Bezüge**

*Politik, Sozialwissenschaften:*

Datenschutz bei Datenbanken,

Probleme von Multi-Userplattformen

**Beispiel: Zootierinformationssystem:**

In einem Zoo leben viele Tierarten, die in unterschiedlichen Gehegen mit und ohne festen Häusern

<p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>• Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln</li> </ul> <p>(c) ©Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> <li>• Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation</li> </ul> <p>(d) Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</p>	<ul style="list-style-type: none"> <li>• bestimmen Primär- und Sekundärschlüssel (M),</li> <li>• modifizieren eine Datenbankmodellierung (M),</li> <li>• ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (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 vorgegebene Normalformen (M),</li> <li>• analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>• verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>• ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> </ul>	<p>untergebracht sind. Die Tiere werden einzeln oder in Gruppen von Pflegern betreut und benötigen jeweils spezielles Futter, welches von verschiedenen Lieferanten bezogen werden kann.</p> <p><b>Beispiel: Buchungssystem</b></p> <p>In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden.</p> <p><b>Beispiel: Fahrradverleih</b></p> <p>Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von <i>BTR</i> können CityBikes, Treckingräder und Mountainbikes ausleihen.</p>
<p><b>2. Nutzung von relationalen Datenbanken</b></p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> <li>• Entwicklung von Fragestellungen zur vorhandenen Datenbank</li> <li>• Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema</li> </ul> <p>(b) SQL-Abfragen</p>		<p><b>außerschulische Lernorte:</b></p> <p>keine</p> <p><b>Formen der Kompetenzüberprüfung:</b></p>

<ul style="list-style-type: none"> <li>• 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 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)</li> </ul> <p>(c) ©Vertiefung an einem weiteren Datenbankbeispiel</p>		S.O.
---	--	------

## Unterrichtsvorhaben V (Q1)

**Thema:** Sicherheit und Datenschutz in Netzstrukturen

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

**Inhaltsfelder:** IS, IMG

**Zeitbedarf:** 12 Unterrichtsstunden

**Inhaltsbezogene Kompetenzen**

**Prozessbezogenen Kompetenzen (Schwerpunkte)**

**Vereinbarungen der Fachkonferenz**

Die Thematik „Netzwerke“ und „Datensicherheit“ schließt sich unmittelbar an das vorhergehende Unterrichtsvorhaben zum Thema Datenbanken an. Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell werden erarbeitet und kennengelernt und miteinander verglichen. Datenbankzugriff aus dem Netz sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren werden analysiert und erläutert. Fallbeispiele zur Datenschutzproblematik und

zum Urheberrecht runden das Unterrichtsvorhaben ab.

<p><b>1. Daten in Netzwerken</b></p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p><b>2. Verschlüsselung und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</b></p> <p>(a) Vertraulichkeit, Integrität, Authentizität in Netzwerken</p> <p>(b) Symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden um Daten im Netz verschlüsselt zu übertragen</p> <p><b>3. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</b></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>• analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A),</li> <li>• untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen 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> <li>• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).</li> </ul>	<p><b>fachübergreifende Bezüge</b></p> <p><i>Mathematik:</i> Primzahlzerlegung und ihre Anwendung bei asymmetrischen Verfahren</p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>
--	---	--

## Unterrichtsvorhaben VI (Q2)

**Thema:** Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

**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?*

**Inhaltsfelder:** DS, IL

**Zeitbedarf:** 24 Unterrichtsstunden

**Inhaltsbezogene Kompetenzen**

**Prozessbezogenen Kompetenzen (Schwerpunkte)**

**Vereinbarungen der Fachkonferenz**

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 (Details siehe Materialien für das Zentralabitur in NRW) 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 Baum Inhalte 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 BinarySearchTree (Details siehe Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

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

<p><b>1. Analyse von Baumstrukturen in verschiedenen Kontexten</b></p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p> <p><b>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p> <p><b>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</b></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung einer abstrakten Klasse Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der An-</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>• implementieren iterative und rekursive Algorithmen</li> </ul>	<p><b>fachübergreifende Bezüge</b></p> <p><b>Beispiel: Ternbaum</b></p> <p>Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p><b>Beispiel: Ahnenbaum</b></p> <p>Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><b>Beispiel: Suchbäume</b> (zur sortierten Speicherung von Daten)</p> <p>Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p><b>Beispiel: Entscheidungsbäume</b></p> <p>Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum.</p> <p><b>Beispiel: Codierungs-</b></p>
--	---	---



<p>wendung inklusive einer sortierten Ausgabe des Baums</p> <p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>	<p>men auch unter Verwendung von dynamischen Datenstrukturen (I),</p> <ul style="list-style-type: none"> <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> <li>• stellen lineare und nicht-lineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<p><b>bäume</b> (Morsecodierer, Huffmannbaum)</p> <p>Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens</p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>
--	---	---

## Unterrichtsvorhaben VII (Q2)

**Thema:** Endliche Automaten und formale Sprachen

**Leitfragen:** *Wie kann man (endliche) Automaten genau beschreiben? Wie können endliche*

*Automaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, endlichen Automaten und regulären Grammatiken?*

**Inhaltsfelder:** FSA

**Zeitbedarf:** 20 Unterrichtsstunden

Inhaltsbezogene Kompetenzen	Prozessbezogenen Kompetenzen (Schwerpunkte)	Vereinbarungen der Fachkonferenz
<p>Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.</p>		
<p>Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet und ihre Erweiterungen über Kellermaschinen zur Turingmaschine analysiert. Parallel dazu werden andere Grammatiken untersucht oder entwickelt und die Chomsky-Hierarchie der formalen Sprachen thematisiert.</p>		
<p><b>1. Endliche Automaten</b></p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p> <p><b>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</b></p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikati-</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),</li> <li>• analysieren und erläutern Grammatiken regulärer Sprachen (A),</li> <li>• zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),</li> <li>• ermitteln die formale</li> </ul>	<p><b>fachübergreifende Bezüge</b></p> <p><i>Sozialwissenschaften:</i> Automaten und Automati- on in der Arbeitswelt</p> <p><b>Beispiele: Automaten im Alltag</b></p> <p>Cola-Automat, Geldspiel- automat, Roboter, Zustandsände- rung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teil- wörter in längeren Zei- chenketten, Akzeptor für</p>

<p>on und Entwicklung von Grammatiken</p> <p>(c) ©Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen, die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p> <p><b>3. Grenzen endlicher Automaten und ihre Erweiterungen</b></p> <p>(a) Beweis, dass ein endlicher Automat nicht in der Lage ist, beliebige, korrekt geklammerte arithmetische Ausdrücke zu erkennen</p> <p>(b) Erweiterung durch einen Zwischenspeicher in Form eines Stapels („Keller“) führt zu Kellerautomaten, die Verwendung von prinzipiell beliebig beschreib- und lesbarem Speicher zur Turingmaschine</p> <p>(c) ©Zunahme der Komplexität formaler Sprachen mit den Möglichkeiten der Maschinen, Chomsky-Hierarchie formaler Sprachen</p> <p>(c) ©</p>	<p>Sprache, die durch eine Grammatik erzeugt wird (A),</p> <ul style="list-style-type: none"> <li>• entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>• entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),</li> <li>• entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),</li> <li>• entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),</li> <li>• modifizieren Grammatiken regulärer Sprachen (M),</li> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),</li> <li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),</li> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).</li> <li>• analysieren und erläutern die Möglichkeiten, endliche Automaten durch Speicher zu erweitern und so erkannte</li> </ul>	<p>Terme</p> <p><b>Beispiele:</b> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliedergrammatik</p> <p><b>Beispiele:</b> Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p> <p><b>außerschulische Lernorte:</b> keine</p> <p><b>Formen der Kompetenzüberprüfung:</b> s.o.</p>
---	--	--

	Grenzen zu überwinden (A)	
--	------------------------------	--

<b>Unterrichtsvorhaben VIII (Q2)</b>		
<b>Thema:</b> Grenzen eines Computers und Grenzen der Automatisierbarkeit		
<b>Leitfragen:</b> <i>Worin liegen die Grenzen von Automaten wie Computern? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?</i>		
<b>Inhaltsfelder:</b> IS, FSA, IMG		
<b>Zeitbedarf:</b> 12 Unterrichtsstunden		
<b>Inhaltsbezogene Kompetenzen</b>	<b>Prozessbezogenen Kompetenzen (Schwerpunkte)</b>	<b>Vereinbarungen der Fachkonferenz</b>
In Abhängigkeit von der Kurszusammensetzung und den Wahlen von Informatik als 3. Abiturfach kann der Punkt 2. zugunsten einer intensiven Wiederholung von Inhalten der Q1 und Vorbereitung auf die schriftliche Prüfung gekürzt werden oder ganz entfallen.		
<p><b>1. Grenzen der Automatisierbarkeit</b></p> <p>(a) Vorstellung des Halteproblems</p> <p>(b) Unlösbarkeit des Halteproblems</p> <p>(c) ©Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p> <p><b>2. Grenzen der Berechenbarkeit</b></p> <p>(d) Unberechenbare Funktionen („Busy Beaver“)</p> <p>(e) Praktikable und impraktikable Probleme (Königsberger Brückenproblem, Problem des Handlungsreisenden, Erfüllbarkeitsproblem)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• untersuchen und beurteilen Grenzen von Computersystemen (A)</li> <li>• analysieren und beurteilen die Grenzen der Einsetzbarkeit von Rechnern und der Problemlösungsmöglichkeiten durch Algorithmen (A)</li> <li>•</li> </ul>	<p><b>fachübergreifende Bezüge</b></p> <p><i>Politik, Gesellschaftswissenschaften</i></p> <p><b>außerschulische Lernorte:</b></p> <p>keine</p> <p><b>Formen der Kompetenzüberprüfung:</b></p> <p>s.o.</p>

## 2.6 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

An der Hildegardis-Schule ist – historisch bedingt – der Anteil der Mädchen immer noch höher als der Anteil der Jungen. Gerade Mädchen haben immer noch mehr Berührungspunkte und eine geringere Affinität zu eher technisch orientierten Fächern wie z.B. Physik, Chemie und eben Informatik. Diese Zurückhaltung abzubauen, gerade Mädchen in ihren Möglichkeiten und Stärken zu fördern und das traditionelle Rollenverhalten zu hinterfragen und zu überwinden ist auch ein Anliegen des Faches Informatik. Ihm versuchen die Fachlehrer im Unterricht durch bewusste Aufmerksamkeit und Eingehen auf die teilweise anderen Zugangswege zu Inhalten und Lösungsstrategien Rechnung zu tragen. Demgemäß sollen die nachstehenden überfachlichen und fachlichen Grundsätze im Unterricht beachtet werden:

### *Überfachliche Grundsätze:*

- Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler und insbesondere auch der Schülerinnen.
- Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- Medien und Arbeitsmittel sind schülernah gewählt.
- Die Schüler und Schülerinnen erreichen einen Lernzuwachs.
- Der Unterricht fördert eine aktive Teilnahme besonders auch der Schülerinnen.
- Der Unterricht fördert die Zusammenarbeit zwischen den Lernenden und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler und Schülerinnen.
- Schüler wie Schülerinnen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- Es herrscht ein positives pädagogisches Klima im Unterricht.

### *Fachliche Grundsätze:*

- Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für Schülerinnen und Schüler an Bedeutsamkeit.
- Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.

- Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

## 2.7 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

### **Allgemeines:**

Die nachfolgende Darstellung enthält auch die Bewertungskriterien für ein Kursangebot Informatik im WP II der Sekundarstufe I. Dies galt bis zur Einführung von „IFO – For-schen mit Informatik“ als fächerübergreifender Versuch und würde auch wieder in Kraft treten, falls der Versuch eingestellt und das Fach Informatik wieder im WP II angeboten würde.

Im Fach Informatik stützt sich die Bewertung der Leistungen auf drei Säulen:

- 1.) Die Leistungen bei der sonstigen Mitarbeit, die alle Aspekte umfasst, wie sie in den Kernlehrplänen aller Fächer dargestellt sind
- 2.) Die Leistungen in der praktischen Arbeit, d.h., der Umsetzung der theoretischen Arbeit am Computer z.B. in Form von Programmen in den verschiedenen Programmiersprachen, Simulationen, Umgang mit Tabellenkalkulationsaufgaben und Datenbanken, Erstellen von Internetseiten unter Einbeziehung von Scriptsprachen, Umsetzung von Modellen zu Automaten usw.
- 3.) Die Leistungen in den schriftlichen Arbeiten (Sek I ) bzw. den Klausuren ( Sek II )

Die Gewichtung der einzelnen Säulen bei der Ermittlung der Halbjahres- bzw. Schuljahresnote richtet sich auch nach ihrem Anteil am Unterricht.

In der Sekundarstufe I sollte die Anteile der sonstigen Mitarbeit und der praktischen Arbeit etwa gleich stark im Unterricht und bei der Leistungsbewertung berücksichtigt werden, so dass in etwa folgende prozentuale Gewichtung anzustreben ist:

sonstige Mitarbeit: 30%

praktische Arbeit: 30%

schriftliche Arbeit: 40%

In der Sek II wird in der Einführungsphase nur eine Klausur pro Halbjahr geschrieben. Die Gewichtung der 3 Beurteilungssäulen sollte daher etwa gleichgewichtig sein. In den folgenden Halbjahren sollte sich die Gewichtung je nach unterrichtlichem Schwerpunkt an der Gewichtung in der Sek II orientieren. In jedem Fall sind die Schülerinnen und Schüler zu Beginn eines Schuljahres - bei deutlicher Akzentverschiebung auch zu Beginn des 2. Halbjahres - über die geplante Bewertungsgrundlage zu unterrichten.

### **Aufgabenstellung und Bewertung von Klassenarbeiten und Klausuren**

#### a) Grundsätzliches

Die **Formulierung der Aufgaben** und die damit verbundenen Anforderungsbereiche richten sich selbstverständlich nach dem Entwicklungsstand der Schüler, dem jeweils

behandelten Unterrichtsgegenstand und evtl. den technischen Gegebenheiten. Gerade im Fach Informatik sollten auch Möglichkeiten berücksichtigt werden, die Arbeit in Teilen oder vollständig am Rechner zu lösen.

Die **Bewertung** richtet sich grundsätzlich nach den Vorgaben von §48 des Schulgesetzes in der gültigen Fassung bzw. den Festlegungen gemäß Abschnitt 4.2. der Richtlinien für das Fach Informatik. Der Schwierigkeitsgrad bzw. die Anforderungsbereiche und die Punktwertung ist dabei so zu wählen, „dass Schülerinnen und Schüler durch Leistungen mit vorwiegend wiederholendem Charakter eine ausreichende Bewertung erzielen können“.[ RiLi S. 77]. Eine gute Klausur hat die Anforderungsbereiche I (Wiedergabe von Kenntnissen) und II ( Anwenden von Kenntnissen ) erfolgreich bewältigt. Aufgabenteile mit erhöhtem selbstständigem Lösungsanteil (Anforderungsbereich III: Problemlösen und Werten) dienen dazu, zwischen einer guten bzw. sehr guten Leistung zu unterscheiden.

#### b) Festlegung von Notengrenzen

Der Erwartungshorizont zu einer schriftlichen Arbeit spiegelt sich in einer Punktwertung wieder, die die o.g. Grundsätze beachtet, gleichermaßen den zeitlichen wie intellektuellen Aufwand bei der Lösung berücksichtigt und Schülern wie Eltern gleichermaßen die Stärken und Schwächen und die daraus resultierende Gesamtnote nachvollziehbar macht.

Da die jeweils zu erreichende Gesamtpunktzahl variieren kann, orientiert sich die Fachschaft Informatik analog zum Fach Mathematik an einen Prozentschlüssel zur Festlegung der Endnote. Dieser Schlüssel variiert je nach Stufe:

In der Sekundarstufe I gilt:

Prozent-raster	100% - 90%	89% - 80%	79% - 65%	64% - 50%	49% - 25%	< 25%
Notenstufe	1 (sehr gut)	2 (gut)	3 (befriedigend)	4 (ausreichend)	5 (mangelhaft)	6 (ungenügend)

An den Rändern des Prozentrasters können zusätzliche Tendenzen (plus / minus ) mit der Note vergeben werden, die keinen rechtlich bindenden Charakter haben, sondern lediglich der Information dienen, wie die erzielte Leistung im doch recht breiten Notenbereich einzuordnen ist.

In der **Sekundarstufe II** wird zwischen der Einführungsphase und der Qualifikationsphase unterschieden. In der Einführungsphase nähern sich die Rastergrenzen den Vorgaben der Qualifikationsphase an. Da auf den halbjahres- und Abschlusszeugnissen der Einführungsphase jedoch noch keine Tendenzen unterschieden werden, was sich insbesondere bei der Notentendenz „ausreichend minus“ auswirkt, legt die Fachschaft Informatik folgendes Prozentraster fest:



**Einführungsphase:**

Grenzen	100-97	96-92	91-87	86-83	82-78	77-74	73-69	68-64	63-60	59-55	54-50	49-45	44-39	38-31	30-24	<24
Note	1 +	1	1 -	2 +	2	2 -	3 +	3	3 -	4 +	4	4 -	5 +	5	5 -	6

**Qualifikationsphase:**

Grenzen	100-95	94-90	89-85	84-80	79-75	74-70	69-65	64-60	59-55	54-50	49-45	44-40	39-32	31-27	26-20	<20
Note	1 +	1	1 -	2 +	2	2 -	3 +	3	3 -	4 +	4	4 -	5 +	5	5 -	6
Punkte	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Die Grobeinteilung der Bewertungen in der Sekundarstufe II folgt dabei den Vorgaben der APO-GOST in der gültigen Fassung.

**2.8 Lehr- und Lernmittel**

Neben der bereits beschriebenen technischen Ausstattung werden keine weiteren Mittel eingesetzt. Die Fachkonferenz hat sich nach der Erprobung verschiedener Lehrwerke dafür entschieden, dort, wo es notwendig und geboten erscheint, entsprechende Materialien selbst zu erstellen und für die Schüler bereit zu stellen. Die erprobten Lehrwerke hatten nach Ansicht der Fachkonferenz stets nur punktuell ihre Stärken. Das genügte der Fachkonferenz nicht, sich für ein durchgängig eingesetztes Lehrwerk zu entscheiden. Darüber hinaus änderten sich die Vorgaben des Ministeriums in den letzten Jahren in so kurzen zeitlichen Abständen, dass die Aktualität eines Lehrwerks nicht für einen vertretbaren Zeitraum gewährleistet war.

**3. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen**

Wie die Übersicht der einzelnen Unterrichtsinhalte aufzeigt, gibt es zahlreiche Möglichkeiten zu fächer- und themenübergreifendem Unterricht. Dieses Potential zu heben ist eine Aufgabe des Koordinators für Schulentwicklung. Eine kontinuierliche, fächerübergreifende Arbeit setzt allerdings eine gewisse Kontinuität im Wahlverhalten und in der Einrichtung eines Kurses auch in der Qualifikationsphase voraus. Das wurde bereits im 1. Kapitel ausführlich erläutert.

Neben dem schulinternen, fächerübergreifenden Unterricht ist auch eine Kooperation mit außerschulischen Partnern wünschens- und erstrebenswert. Angebote oder Anfragen der TU Dortmund oder von Firmen, die durch ehemalige Schülerinnen und Schüler auf die Hildegardis-Schule aufmerksam gemacht wurden, liegen zwar vor, konnten aber wegen der fehlenden Kurse nicht weiterverfolgt, eingerichtet oder vertieft werden. Die Fachschaft Informatik ist sich darin einig, dass derartige Angebote zur Attraktivitätssteigerung

des Faches einerseits, aber auch für die Außendarstellung der Schule ein Gewinn sein könnten.

#### **4. Qualitätssicherung und Evaluation**

Neben den obligatorischen Fachkonferenzen findet ein regelmäßiger und kontinuierlicher Austausch der Fachkollegen untereinander über Unterrichtsmodelle, gelungene Unterrichtsreihen oder Fortbildungsmöglichkeiten statt. Die Klausuren werden standardmäßig untereinander ausgetauscht, so dass jederzeit die Information über Inhalte Anforderungen und Beurteilungskriterien gegeben ist. Darüber hinaus ist an der Hildegardis-Schule eine obligatorische Evaluation des Unterrichts und des Unterrichtenden durch die Schüler zum Ende eines Halbjahres vorgesehen. Die bisherigen Rückmeldungen lassen keinen besonderen Handlungsbedarf über eine verstärkte Werbung für das Fach hinaus erkennen.

