

# ***Gymnasium Holthausen, Hattingen***

**Schulinterner Lehrplan  
zum Kernlehrplan für die gymnasiale Oberstufe  
für das Fach**

## **Informatik ( SII )**

**unter Berücksichtigung der Abiturvorgaben  
2019 und 2020**

**(Stand: 01.08.2018)**

***Zusammengestellt durch D. von Horn und D. Szebrowski.  
Letzte Überarbeitung am 30.04.2018 durch D. Szebrowski.***

# Inhalt

	Seite
<b>1 Die Fachgruppe Informatik am Gymnasium Holthausen, Hattingen</b>	<b>3</b>
<b>2 Entscheidungen zum Unterricht</b>	<b>5</b>
2.1 Unterrichtsvorhaben	5
2.2 Übersichtsraster zu den Unterrichtsvorhaben und deren Konkretisierung	6
I) Einführungsphase EF (10)	7
II) Qualifikationsphase Q1 (11)	24
III) Qualifikationsphase Q2 (12)	39
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	48
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	49
<b>3 Qualitätssicherung und Evaluation</b>	<b>51</b>

## **1 Die Fachgruppe Informatik am Gymnasium Holthausen in Hattingen**

Beim Gymnasium Holthausen (im Folgenden kurz: GyHo) handelt es sich um eine vierzügige Schule am Ortsrand von Hattingen im Ortsteil Holthausen mit zurzeit ca. 900 Schülerinnen und Schülern und ca. 65 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst den im Wesentlichen die Ortsteile Holthausen, Blankenstein, Welper und Stadtmitte sowie umliegende Ortschaften (z.B. Teile von Sprockhövel). Im Bereich der Sekundarstufe II finden einige Kurse in Kooperation mit dem Gymnasium an der Waldstraße in Hattingen statt.

Das Fach Informatik wird am GyHo in den Jahrgangsstufen 8 und 9 im Wahlpflichtbereich II (WP II) jeweils in einem Kurs mit ca. 24 Schülerinnen und Schülern zweistündig unterrichtet. In der zweijährigen Laufzeit dieser Kurse liegt der Schwerpunkt in der altersstufengerechten Betrachtung von Alltagsanwendungen, weshalb der Kurs unter der Bezeichnung „Praktische Informatik“ mit wechselnden Themenschwerpunkten angeboten wird. Der Unterricht erfolgt dabei zum überwiegenden Teil in Form von Projekten, die durchaus auch fächerverbindend sein können.

Organisatorisch ist das Fach Informatik in der Sekundarstufe I in den MINT-Zweig der Schule eingebunden.

In der Sekundarstufe II bietet das GyHo für die Schülerinnen und Schüler in allen Jahrgangsstufen in der Regel mehrere Grundkurse in Informatik an, wobei durch die Kooperation auch Teilnehmer/-innen vom Gymnasium an der Waldstraße hinzu kommen können.

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

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache JAVA durchgeführt. Sowohl in der Einführungs- als auch der Qualifikationsphase kommen didaktische Bibliotheken zum Einsatz (SuM), welche das Erstellen von grafischen Programmen und Benutzeroberflächen und die Anbindung an Netzwerke erleichtern. Bei den Datenstrukturen kommen seit 2016 die im Abitur vorgeschriebenen Klassen mit generischen Datentypen zum Einsatz; auch bei den Datenbanken (und den Graphen) werden diese sog. „Abiturklassen“ verwendet.

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

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik des GyHo aus nur einer Lehrkraft; zwei weitere Lehrkräfte haben zwar ein Zertifikat erworben, jedoch nur wenig Unterrichtserfahrung. Es stehen zwei Computerräume mit jeweils 15 Computerarbeitsplätzen zur Verfügung; alle Arbeitsplätze sind an das schulinterne Rechnernetz angeschlossen (JAMBO), so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der beiden Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

Der Unterricht erfolgt in der SI überwiegend in Doppelstunden von 90 Minuten. Die Kursblockung in der SII sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor.

## 2 Entscheidungen zum Unterricht

### 2.1 Unterrichtsvorhaben

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

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

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer 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. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, 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 Übersichtsraster zu den Unterrichtsvorhaben und deren Konkretisierung

Als Lehrbuch kommt in der Einführungsphase das Lehrbuch von Bernhard Schriek „Informatik mit JAVA“ (Band 1) aus dem Nili-Verlag (Werl) zum Einsatz. In der Qualifikationsphase wird das Lehrwerk „Informatik“ (Band 2) aus dem Schöningh-Verlag verwendet.

In der Einführungsphase werden die Kern- und die Werkzeugklasse der Bibliothek „Stifte und Mäuse“ (SuM) benutzt. Die Installationspakete und Dokumentationen sowie eine lauffähige Programmierumgebung (BlueJ) stehen den Schüler/-innen zum Download im Informatikbereich der Homepage des GyHo zur Verfügung.

In der Qualifikationsphase wird aus der Bibliothek SuM statt der Kern- die Ereignisklasse verwendet sowie zusätzlich die Klassen mit den Bedienungskomponenten, den Netzwerkdiensten als auch der Einbindung medialer Elemente. Die Unterrichtsvorhaben werden stets unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse

<http://www.standardsicherung.schulministerium.nrw.de/cms/zentralabitur-gost/faecher/fach.php?fach=15>  
(abgerufen: 30. 04. 2018)

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

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 (K).

## I) Jahrgangsstufe EF (10)

Einführungsphase EF (10)	
<p><u>Unterrichtsvorhaben E-I</u></p> <p><b>Thema:</b> <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Formale Sprachen und Automaten</li><li>• Algorithmen</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte von vorgegebenen Klassen</li><li>• Syntax und Semantik einer Programmiersprache (JAVA)</li><li>• Elementare Datentypen</li></ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p><b>Thema:</b> <i>Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Implementieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li><li>• Kontrollstrukturen (inkl. Struktogrammform)</li></ul> <p><b>Zeitbedarf:</b> 18 Stunden</p>

## Einführungsphase EF (10)

### Unterrichtsvorhaben E-III

**Thema:**

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

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Daten und ihre Strukturierung
- Algorithmen

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung von Algorithmen

**Zeitbedarf:** 21 Stunden

### Unterrichtsvorhaben E-IV

**Thema:**

*Modellierung und Implementierung von abstrakten Klassen anhand von grafischen Projekten*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Daten und ihre Strukturierung
- Algorithmen

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung von Algorithmen

**Zeitbedarf:** 15 Stunden



## Einführungsphase EF (10)

### Unterrichtsvorhaben E-V

**Thema:**

*Einführung in die ereignisorientierte Programmierung*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Algorithmen
- Daten und ihre Strukturierung

**Inhaltliche Schwerpunkte:**

- Ereigniserkennung, -verteilung, -bearbeitung
- Analyse, Entwurf und Implementierung einfacher Algorithmen

**Zeitbedarf:** 18 Stunden

### Unterrichtsvorhaben E-VI

**Thema:**

*Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes*

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

**Inhaltliche Schwerpunkte:**

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung

**Zeitbedarf:** 9 Stunden

**Summe Einführungsphase: 89**

## **Unterrichtsvorhaben EF-I**

**Thema:** *Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen*

**Leitfrage:** *Wie lässt sich die Realität zum Lösen konkreter Aufgabenstellungen in ein informatorisches Modell abbilden ?*

### **Vorhabenbezogene Konkretisierung:**

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei ist zu berücksichtigen, dass für manche Schülerinnen und Schüler in der Einführungsphase der erste Kontakt mit dem Unterrichtsfach Informatik stattfindet, so dass zu Beginn Grundlagen des Fachs behandelt werden müssen.

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

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektkarten, Klassenkarten oder Beziehungsdiagramme eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe der didaktischen Programmierumgebung SuM/BlueJ begonnen. Die von der Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert und zwischen Deklaration, Initialisierung und Methodenaufrufen unterschieden werden.

Da bei der Umsetzung dieser ersten Projekte konsequent auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

**Zeitbedarf:** 8 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Identifikation von Objekten</b></p> <p>(a) Am Beispiel eines lebensweltnahen 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>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 Besitzbeziehung 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><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• grafische Muster</li> <li>• „Haus vom Nikolaus“ u.ä.</li> </ul> <p><i>Materialien:</i> Buch, Band 1, Kapitel 2 und 3</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 didaktischen Lernumgebungen SuM/BlueJ</p>		

<b>3. Implementierung statischer Grafikprogramme</b>		
--	--	--

(a) Grundaufbau einer Java-Klasse

(b) Deklaration und Initialisierung von Objekten

(c) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung)

## Unterrichtsvorhaben EF-II

**Thema:** Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

**Leitfragen:** *Wie lassen sich Animationen und Simulationen von Grafikobjekten unter Berücksichtigung von Tastatur- und Mauseingaben realisieren?*

### **Vorhabenbezogene Konkretisierung:**

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben des Benutzers gesteuerte Grafikobjekte aufweisen. Zunächst wird ein Projekt bearbeitet, das in Anlehnung an das vorangegangene Unterrichtsvorhaben ermöglicht, Freihandzeichnungen zu erstellen. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt und eingeführt.

Das Unterrichtsvorhaben wird fortgeführt mit einem Projekt, das komplexere grafische Elemente beinhaltet, in denen erstmalig animierte Objekte erstellt werden.

Die Assoziationsbeziehungen zwischen Klassen werden in diesem Unterrichtsvorhaben nicht behandelt. Sie stellen den Schwerpunkt des folgenden Vorhabens dar. Vorbereitend auf das nächste Vorhaben können aber am Ende des Projektes erste eigene Klassen erstellt werden.

**Zeitbedarf:** 18 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Erstellung eines einfachen Malprogramms</b></p> <p>(a) Kontinuierliches Zeichnen mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife</p> <p>(c) Einseitige und mehrseitige Verzweigungen zur Realisierung von komplexeren Animationen (IF- und SWITCH-Anweisungen)</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 zu (M),</li> </ul>	<p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• Freihandzeichnen</li> <li>• Simulation eines Dartspiels</li> </ul> <p><i>Materialien:</i> Buch, Band 1, Kapitel 4 und 5</p>
<p><b>2. Bewegungsanimationen am Beispiel einfacher grafischer Objekte</b></p> <p>(a) Kontinuierliche Verschiebung eines Grafikobjektes mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(c) Berechnung von Abständen mit Hilfsvariablen</p>	<ul style="list-style-type: none"> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• implementieren Algorithmen, Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),</li> </ul>	
	<ul style="list-style-type: none"> <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 EF-III**

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

**Leitfrage:** *Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?*

### **Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau komplexerer Objektbeziehungen. Während in vorangegangenen Unterrichtsvorhaben Objekte nur jeweils solchen Objekten Nachrichten schicken konnten, die sie selbst erstellt haben, soll in diesem Unterrichtsvorhaben diese hierarchische Struktur aufgebrochen werden.

Dazu bedarf es zunächst einer präzisen Unterscheidung zwischen Objektreferenzen und Objekten, so dass klar wird, dass Dienste eines Objektes von unterschiedlichen Objekten über unterschiedliche Referenzen in Anspruch genommen werden können. Auch der Aufbau solcher Objektbeziehungen muss thematisiert werden. Des Weiteren wird das Prinzip der Vererbung im objektorientierten Sinne angesprochen. Dazu werden die wichtigsten Varianten der Vererbung anhand von verschiedenen Projekten vorgestellt. Zunächst wird die Vererbung als Spezialisierung im Sinne einer einfachen Erweiterung einer Oberklasse vorgestellt. Modellierungen sollen in Form von Implementationsdiagrammen erstellt werden.

**Zeitbedarf:** 21 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. 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>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 zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> </ul>	<p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• erweitertes Dart-Projekt</li> <li>• Billiardspiel</li> <li>• Rotkäppchen oder Minigolf</li> </ul> <p><i>Materialien:</i> Buch, Band 1, Kapitel 6 und 7</p>
<p><b>2. 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 ent-</p>	<ul style="list-style-type: none"> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• modifizieren einfache Algorithmen und Pro-</li> </ul>	



sprechenden Animationsmethode	gramme (I),	
<p><b>3. Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten</b></p> <p>(a) Dokumentation der Klassen des Projekts</p> <p>(b) Ergänzung einer Kollisionsabfrage durch zusätzliche Assoziationsbeziehungen in Diagramm, Dokumentation und Quellcode</p> <p>(c) Verallgemeinerung der neuen Verwendung von Objektreferenzen</p> <p>(d) Vertiefung: Entwicklung weiterer Spiele und Simulationen mit vergleichbarer Grundmodellierung</p>	<ul style="list-style-type: none"> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)</li> <li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).</li> </ul>	

## Unterrichtsvorhaben EF-IV

**Thema:** Modellierung und Implementierung von abstrakten Klassen anhand von grafischen Projekten

**Leitfragen:** *Wie können Daten und Dienste durch Generalisierung effizienter gestaltet werden ?*

### Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Verallgemeinerung von Daten mit en zugehörigen Diensten durch Auslagern in eine abstrakte Oberklasse.

Es wird ein Grafikprojekt behandelt, welches das Verständnis von Vererbung um die Aspekte der Polymorphie und der späten Bindung erweitert. Dabei werden die abstrakten Dienste der Oberklasse in den Unterklassen durch entsprechende Methoden konkretisiert.

In einem weiteren Projekt wird bereits die Verkettung von Objekten durch das Nachfolgerkonzept vorbereitend behandelt.

**Zeitbedarf:** 15 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</b> (a) Einführung der Objektselektion mit der Maus (b) Steuerung einfacher grafischer Objekte über eine Referenz <i>aktuell</i> , die jeweils durch eine Klickselektion mit der Maus auf ein neues Objekt gesetzt werden kann	Die Schülerinnen und Schüler <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></ul>	<i>Beispiele:</i> <ul style="list-style-type: none"><li>• Figurprojekt</li><li>• Simulation eines fahrenden Zuges</li></ul> <i>Materialien:</i> Buch, Band 1, Kapitel 8 und 9

	<ul style="list-style-type: none"> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> </ul>	
<p><b>2. Erweiterung des Referenzbegriffes durch Verwenden abstrakter Oberklassen</b></p> <p>(a) Bilden eine gemeinsamen Oberklasse aus gemeinsamen Attributen und Diensten mehrerer ähnlicher Klassen</p> <p>(b) Verwenden einer Referenz vom abstrakten Datentyp (Polymorphie)</p> <p>(c) Anwenden des späten Bindens als Prinzip zur Dynamisierung von Programmabläufen</p>	<ul style="list-style-type: none"> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren abstrakte 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> </ul>	
<p><b>3. Einführung in die Phasen von Softwareentwicklung</b></p> <p>(a) Betrachtung der Techniken eines Modellentwurfs</p> <p>(b) Arbeitsteilige Entwicklung eines Softwareprojektes</p> <p>(c) Zusammenführung und Evaluierung der einzelnen Module</p>		

## **Unterrichtsvorhaben EF-V**

**Thema:** Einführung in die ereignisgesteuerte Programmierung

**Leitfragen:** *Wie lassen sich verschiedene Objekte durch Eingabeereignisse steuern ? Wie können Objekte selbst auf Ereignisse reagieren?*

### **Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich mit der Möglichkeit, Objekte zu Bearbeitern von Ereignissen zu machen, um so die Möglichkeit zu bekommen, interaktiv mit ihnen zu kommunizieren.

Es wird ein Grafikprojekt behandelt, welches die Erkennung, die Verteilung und die Bearbeitung von Maus- und Tastaturereignissen verdeutlicht. Die Objekte reagieren selbst auf Informationen, die sie von der neu zu entwickelnden „Ereignisanwendung“ mitgeteilt bekommen. Dazu werden sowohl die Standardobjekte Bildschirm, Tastatur und Maus als auch die Erkennung von Ereignissen generalisiert und in eine eigene (abstrakte) Klasse ausgelagert. Objekte werden zu „Ereignisbearbeitern“.

In Ergänzung werden am Ende durch Hinzunahme einer weiteren SuM-Bibliothek neue Komponenten zur Gestaltung von Benutzeroberflächen (Knöpfe, Textfelder, Etiketten usw.) eingeführt, die ebenfalls auf Ereignisse reagieren und einen interaktiven Programmablauf ermöglichen.

**Zeitbedarf:** 18 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Entwicklung einer mehrstufigen Ereignisverarbeitung</b></p> <p>(a) Gliederung in Ereigniserkennung, -verteilung und -bearbeitung</p> <p>(b) Entwicklung von Bearbeiterdiensten für Tastatur- und Mausereignisse</p> <p>(c) Auslagerung der Ereignisbearbeitung in die Objekte („Ereignisbearbeiter“)</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> </ul>	<p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• Projekt „Sackgasse“</li> <li>• Dartspiel als Ereignisanwendung</li> </ul> <p><i>Materialien:</i> Buch, Band 1, Kapitel 10 und 11</p>
<p><b>2. Erweiterung der Ereignisverarbeitung durch Generalisierung</b></p> <p>(a) Erstellen einer anwendungsunabhängigen Erkennungsschleife</p> <p>(b) Auslagerung der Ereigniserkennung und der Standardobjekte (Bildschirm, Tastatur, Maus) in die Oberklasse „Ereignisanwendung“</p> <p>(c) Automatisierung der Ereigniserkennung durch Abtrennen des Anwenderspezifischen Programmteils (SuMANwendung)</p>	<ul style="list-style-type: none"> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren abstrakte 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> </ul>	
<p><b>3. Verallgemeinerung der Ereignisanwendung</b></p> <p>(a) Einführung der Bibliothek <code>sum.ereignis</code></p> <p>(b) Knöpfe als Ereignisbearbeiter</p>	<ul style="list-style-type: none"> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)</li> </ul>	

(c) Anmeldung von weiteren Objekten bei der EBAAnwendung		
--	--	--

## Unterrichtsvorhaben EF-VI

**Thema:** Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

**Leitfrage:** *Welche Entwicklung durchlief die moderne Datenverarbeitung und welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus?*

### Vorhabenbezogene Konkretisierung:

Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Schülerinnen und Schüler sollen selbstständig (z.B. in Form von Referaten) informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

**Zeitbedarf:** 9 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</b></p> <p>(a) Mögliche Themen:</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> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informationssystemen (A),</li> <li>erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),</li> <li>stellen ganze Zahlen und Zeichen in Binär-</li> </ul>	<p><i>Materialien:</i></p> <p>Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>

<ul style="list-style-type: none"> <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>codes dar (D),</p> <ul style="list-style-type: none"> <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>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><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz</p>

## II) Jahrgangsstufe Q1 (11)

Qualifikationsphase Q1 (11)	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b> <i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</li><li>• Informatiksysteme</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Analyse, Entwurf und Implementierung von Algorithmen</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Nutzung von Informatiksystemen</li></ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b> <i>Erarbeitung einer weiteren Lösungsmethode für sich wiederholende Problemstellungen</i></p> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Analyse, Entwurf und Implementierung von Algorithmen</li><li>• Algorithmen in ausgewählten informatischen Kontexten</li><li>• Syntax und Semantik einer Programmiersprache</li></ul> <p><b>Zeitbedarf:</b> 9 Stunden</p>



## Qualifikationsphase Q1 (11)

### Unterrichtsvorhaben Q1-III

**Thema:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 18 Stunden

### Unterrichtsvorhaben Q1-IV

**Thema:**

*Suchen und Sortieren auf linearen Datenstrukturen*

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 16 Stunden

## Qualifikationsphase Q1 (11)

### Unterrichtsvorhaben Q1-V

**Thema:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 24 Stunden

**Qualifikationsphase Q1 , Summe: 75**

## Q1-I: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
----------------------	-----------------------------	--------------------------------

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Entwicklung und Testen einer Fachklasse (Model)</b></p> <ul style="list-style-type: none"> <li>• Erstellen eines Implementationsdiagramms für die Klasse</li> <li>• Schreiben einer Dokumentation der Klasse</li> <li>• Wiederholen der Entwicklung von Klassen sowie Get- und Set-Methoden</li> </ul>	<p>Die Schülerinnen und Schüler</p> <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 (M).</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M).</li> <li>• dokumentieren Klassen (D).</li> <li>• implementieren Klassen in einer Programmiersprache (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> </ul>	<p><b>z.B. Bruchrechner-Projekt</b></p> <p>Klasse Bruch</p> <p>Testen als Exemplar der Klasse in der Werkbank</p>
<p><b>2. Entwicklung einer Rechenklasse</b></p> <ul style="list-style-type: none"> <li>• Kommunikation zwischen Klassen</li> <li>• Entwicklung von Algorithmen</li> </ul>	<ul style="list-style-type: none"> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D).</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (D).</li> <li>• modellieren Klassen mit ihren</li> </ul>	<p>Klasse Bruchrechner</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
	Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M).	
<b>3. Entwicklung einer Oberflächenklasse (Control, View)</b>	s.o.	
<b>4. MVC-Konzept</b>	analysieren und erläutern objektorientierte Modellierungen (A).	Erläutern des Konzepts anhand des Beispiels

## Q1-II: Erarbeitung einer weiteren Lösungsmethode für sich wiederholende Problemstellungen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p><b>Rekursion als algorithmische Lösungsmethode</b></p> <ul style="list-style-type: none"> <li>• Herausarbeitung der Merkmale rekursiver Lösungen</li> <li>• Grafische Darstellung u.a. als Struktogramm</li> <li>• Vergleich Iteration/Rekursion mit Bewertung</li> </ul>	<p>Die Schülerinnen und Schüler</p> <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 (M).</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M).</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>• testen Programme systematisch anhand von Beispielen</li> </ul>	<p><b>Beispiele:</b></p> <ul style="list-style-type: none"> <li>• Fibonacci</li> <li>• Fakultät</li> <li>• Türme von Hanoi</li> <li>• Labyrinth (als Beispiel für Backtracking)</li> </ul>

## Q1-III: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p><b>1. Die (generische) Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</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>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 Programme (I).</li> <li>• implementieren <u>iterative</u> <u>und</u> <u>rekursive</u> Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I).</li> <li>• nutzen die Syntax und Semantik</li> </ul>	<p><i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger)</p> <p>Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird.</p> <p>Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse Queue.</p> <p>Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
	<p>einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I).</p> <ul style="list-style-type: none"> <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>sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.</p>
<p><b>2. Die (generische) Datenstruktur Stapel im Anwendungskontext unter Nutzung 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 Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>	<ul style="list-style-type: none"> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	
<p><b>3. Die (generische) Datenstruktur lineare 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><i>Beispiel: Abfahrtslauf</i></p> <p>Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p> <p><u>Oder:</u> Beispiel <i>Musiker</i></p>



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p><b>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</b></p>		<p><i>Beispiel:</i> Skispringen  Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p>

## Q1-IV: Suchen und Sortieren auf linearen Datenstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<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 (ggf. aber auch iterativ)</p> <p>(c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</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 die Syntax und Semantik einer Programmiersprache bei der</li> </ul>	<p><b>Projekt: 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>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</b></p> <p>(a) Entwicklung und Implementierung eines einfachen</p>		<p><i>Beispiel:</i> Bundesjugendspiele (s.o.)</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele und Materialien
<p>Sortierverfahrens für eine Liste            (b) Entwicklung eines rekursiven Sortierverfahren für eine Liste (Quicksort)</p>	<p>Implementierung und zur Analyse von Programmen (I).</p> <ul style="list-style-type: none"> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	
<p><b>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</b>            (a) Grafische Veranschaulichung der Sortierverfahren            (b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren            (c) Beurteilung der Effizienz der beiden Sortierverfahren</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> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	<p><i>Beispiel:</i> Bundesjugendspiele</p>

## Q1-V: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<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>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</li> </ul>	<p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten, entwickelt aus dem Verfahren der binären Suche)</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><i>Beispiel:</i> Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht</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><i>Beispiel:</i> Ternbaum</p>

	<p>Möglichkeiten der Polymorphie (M).</p> <ul style="list-style-type: none"> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M).</li> <li>• implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I).</li> <li>• modifizieren Algorithmen und Programme (I).</li> </ul>	<p>Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p><i>Beispiel:</i> Entscheidungsbäume 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>2. Die (generische) 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>	<ul style="list-style-type: none"> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I).</li> <li>• 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> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	<p><i>Beispiele:</i> s. o.</p> <p>Durchlaufarten: Suchbaum, Termbaum In-Order Termbaum in UPN: Post-Ordner</p>

<p><b>3. Die (generische) 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 des Interface ComparableContent</p> <p>(d) zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(e) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p><i>Beispiel:</i> Informatikerbaum als Suchbaum  In einem binären Suchbaum werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> <li>• Einfügen der Informatiker-Daten in den Baum</li> <li>• Suchen nach einem Informatiker über den Schlüssel Name</li> <li>• Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge</li> </ul>
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>		<p><i>Beispiel:</i> Huffman-Codierung  Aufgeteilt in mehrere Teilprojekte mit BinaryTree und BinarySearchTree</p>

### III) Jahrgangsstufe Q2 (12)

Qualifikationsphase Q2 (12)	
<p><u>Unterrichtsvorhaben Q2-I</u></p> <p><b>Thema:</b> <i>Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</i></p> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</li><li>• Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Datenbanken</li><li>• Algorithmen in ausgewählten informatischen Kontexten</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Sicherheit</li></ul> <p><b>Zeitbedarf:</b> 20 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p><b>Thema:</b> <i>Endliche Automaten und formale Sprachen</i></p> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Endliche Automaten und formale Sprachen</li><li>• Algorithmen</li></ul> <p><b>Zeitbedarf:</b> 18 Stunden</p>

## Qualifikationsphase Q2 (12)

### Unterrichtsvorhaben Q2-III

**Thema:**

*Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit*

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

**Zeitbedarf:** 9 Stunden

### Unterrichtsvorhaben Q2-IV

**Thema:**

*Sicherheit und Datenschutz in Netzstrukturen*

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 6 Stunden



## Qualifikationsphase Q2 (12)

### Unterrichtsvorhaben Q2-V

**Thema:**

*Wiederholung und Vertiefung  
ausgewählter Kompetenzen und Inhalte  
der Qualifikationsphase*

**Inhaltsfelder:**

(themen- und stufenübergreifend)

**Inhaltliche Schwerpunkte:**

Programmierprojekt

- Modellierung
- Datenstrukturen
- Klassenentwurf
- Klassenbeziehungen
- Lösungsalgorithmen
- Programmieren und Testen
- Dokumentation und Präsentation

**Zeitbedarf:** 15 Stunden

**Qualifikationsphase Q2 , Summe: 68**

## Q2-I: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. 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> <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: =,</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A).</li> <li>• analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A).</li> <li>• analysieren und erläutern eine Datenbankmodellierung (A).</li> <li>• erläutern die Eigenschaften normalisierter Datenbankschemata (A).</li> <li>• bestimmen Primär- und Sekundärschlüssel (M).</li> <li>• ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M).</li> <li>• modifizieren eine Datenbankmodellierung (M).</li> <li>• modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M).</li> <li>• überführen Datenbankschemata in die 1. bis 3. Normalform (M).</li> <li>• verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I).</li> </ul>	<p><i>Beispiel: VideoCenter</i></p> <p>VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren. Unter <a href="http://dokumentation.videocenter.schule.de">http://dokumentation.videocenter.schule.de</a> (abgerufen: 30. 03. 2014) findet man den Link zu dem VideoCenter-System sowie nähere Informationen. Lesenswert ist auch die dort verlinkte „Dokumentation der Fallstudie“ mit didaktischem Material, welches alternativ bzw. ergänzend zu der im Folgenden beschriebenen Durchführung verwendet werden kann.</p>

---

<p>&lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL)</p>	<ul style="list-style-type: none"><li>• ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D).</li><li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D).</li><li>• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li></ul>	
--	---	--

## Q2-II: Endliche Automaten und formale Sprachen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<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, deterministischer Automaten</p> <p>(c) Simulation eines Akzeptors in der Programmiersprache</p> <p>(d) Transformation eines nichtdeterministischen Automaten in einen deterministischen Automaten</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 Sprache, die durch eine Grammatik erzeugt wird (A).</li> </ul>	<p>Als roter Faden der Reihe dient die Idee, einen „Taschenrechner“ für Terme zu entwickeln.</p> <p><i>Beispiele:</i> Eingabe eines Taschenrechners, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Cola-Automat, Geldspielautomat Zahlakzeptor in Java</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, Modifikation 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>(e) Produktionen mit <math>\epsilon</math> (Epsilon)</p>	<ul style="list-style-type: none"> <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</li> </ul>	<p><i>Beispiele:</i> reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen repräsentieren, Satzgliedergrammatik</p>

<p><b>3. Grenzen endlicher Automaten</b></p>	<p>erzeugt (M).</p> <ul style="list-style-type: none"> <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>• transformieren nicht-deterministischer endlicher Automaten in deterministische endliche Automaten (D)</li> </ul>	<p><i>Beispiele:</i> Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p>
--	---	---

## Q2-III: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</b></p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<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-Neumann-Architektur“ (A).</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<p><i>Beispiel:</i> Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell</p> <p><i>Materialien:</i> Verwenden eines Von-Neumann-Simulators, z. B: Rechner-Baukasten, MIPS-Generator</p>
<p><b>2. 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><i>Beispiel:</i> Halteproblem</p>

## Q2-IV: Sicherheit und Datenschutz in Netzstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</b></p> <p>(a) Beschreibung eines Webserverzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise einer Website-Abfrage</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <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> </ul>	<p><i>Materialien:</i>            Protokolle: HTTP, HTTPS, POP3            Dabei Analyse eines fertigen POP3-Clients</p>
<p><b>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</b></p>		<p><i>Materialien:</i>            Materialblatt Datenschutzgesetz</p>

---

## Unterrichtsvorhaben Q2-V:

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte der gesamten Qualifikationsphase an Hand eines Programmierprojektes.

### 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Gymnasiums Holthausen die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

#### Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schüler/innen erreichen einen Lernzuwachs.
- 6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
- 7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
- 9) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- 10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- 11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- 12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- 13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 14) Es herrscht ein positives pädagogisches Klima im Unterricht.

#### Fachliche Grundsätze:

- 15) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 16) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- 17) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 18) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- 19) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 20) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 21) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.



---

## 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

**Hinweis:** Sowohl die Schaffung von **Transparenz bei Bewertungen** als auch die Vergleichbarkeit von Leistungen sind das Ziel, innerhalb der gegebenen Freiräume Vereinbarungen zu Bewertungskriterien und deren Gewichtung zu treffen.

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Gymnasiums Holthausen im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

### 2.3.1 Beurteilungsbereich Klausuren

#### **Verbindliche Absprachen:**

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

#### **Instrumente:**

- Einführungsphase: 1 Klausur je Halbjahr  
Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr  
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Gemäß dem Beschluss der Lehrerkonferenz kann in Q 1.2 die erste Klausur durch eine Facharbeit ersetzt werden.

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

#### **Kriterien**

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die in der Summe den einzelnen Notenstufen zugeordnet werden.

Von diesem System kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOST §13 (2)) angemessen erscheint.

Die Note ausreichend (5 Punkte) soll im Normalfall bei Erreichen von 40-45 % der Hilfspunkte erteilt werden.

---

## 2.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

### Leistungsaspekte

#### Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen

#### Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

#### Sonstige schriftliche Leistungen

- Arbeitsmappe oder Dokumentation zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen von ca. 20-25 Minuten Dauer, die den Stoff der letzten ca. 4–6 Stunden umfassen
- Bearbeitung von schriftlichen Aufgaben im Unterricht

### Kriterien

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

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

---

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

### **Grundsätze der Leistungsrückmeldung und Beratung**

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen

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

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

Leistungsrückmeldungen erfolgen auch in der Einführungsphase im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.

## **3 Qualitätssicherung und Evaluation**

Das schulinterne Curriculum stellt keine starre Größe dar, sondern ist als „lebendes Dokument“ zu betrachten. Dementsprechend sind die Inhalte stetig zu überprüfen, um ggf. Modifikationen vornehmen zu können. Die Fachkonferenz (als professionelle Lerngemeinschaft) trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches bei.

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.

Das schulinterne Curriculum (siehe 2.1) ist ab sofort verbindlich und wird in der Folgezeit kontinuierlich weiter entwickelt.