

# **Schulinterner Lehrplan Informatik Sek II**

## **1 Die Fachgruppe Informatik des Couven-Gymnasiums**

Das Couven-Gymnasium ist eine fünfzügige Schule mit derzeit ca. 1200 Schülerinnen und Schülern der Stadt Aachen.

Am Couven-Gymnasium wird von der Klasse 5 bis 7.1 durchgängig in allen Klassen ITG (Informationstechnologische Grundlagen) unterrichtet. In diesem Rahmen erhalten die Schülerinnen und Schüler des Couven-Gymnasiums in den Jahrgangsstufen 5 und 6 einen Einblick in die grundlegende Funktionsweise von Rechner- und Dateisystemen, erlernen eine Textverarbeitung, erkennen und diskutieren Gefahren im Internet und lernen mit Vortragssoftware umzugehen sowie gute Präsentationen zu erstellen. Als Vorbereitung dazu dient das Erlernen einer gezielten Internetrecherche. Die Präsentationen werden, wenn möglich, in Kombination mit anderen Fächern (z.B. Biologie, Geschichte) durchgeführt. Darüber hinaus erhalten sie einen Einblick in Bild- und Filmbearbeitung. In der 7.1. erhalten die SuS einen Einblick in das fächerübergreifende Arbeiten zwischen Informatik, Physik und Technik, indem sie einen Microcontroller löten, programmieren und ihr Tun reflektieren. Das Fach Informatik wird dann ab der Jahrgangsstufe 8 im Wahlpflichtbereich II unterrichtet. Ca. ein Viertel der Schülerinnen und Schüler besucht diesen zweijährigen Kurs, in dem in altersstufengerechter Weise unter anderem die Grundlagen der Algorithmik am Beispiel verschiedener didaktischer Lernumgebungen (z.B. Scratch, Android App-Inventor, Logo) sowie die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen (z.B. Arduino, Digitalsimulator) thematisiert werden. Die Unterteilung von Rechnernetzen in Client und Server wird im Rahmen der Webseiten-Erstellung mit HTML, CSS, php und Java-Script thematisiert. Ein kurzer Ausblick auf die Anbindung an Datenbanken gibt eine gute Vorbereitung auf die Oberstufen-Informatik. Die Programmierung von NXT/EV3-Lego-Robotern führt dann zur Anwendung einer konkreten, aber noch vereinfachten Programmiersprache.

Regelmäßig in Kooperation mit der RWTH, aber auch intern durch Schülerinnen aus dem WP-Informatik-Bereich der Klasse 9, durchgeführte zweitägige Roboter-Workshops bieten schon früh einen Einblick in das Fach Informatik. Teilweise werden die Workshops nur für Mädchen durchgeführt. Somit konnte am Couven die Mädchenquote seit 2009 in Informatik erhöht werden.

Im WP-Informatik-Bereich der Sekundarstufe I gibt es normalerweise seit einigen Jahren einen gesonderten Mädchenkurs. Die Mädchen erwerben neben den normalen curricularen Unterrichtsinhalten zusätzlich Kompetenzen zur Durchführung von Roboter-Workshops für jüngere Schüler und Schülerinnen am Couven.

Insofern verfügen alle Schülerinnen und Schüler unabhängig von der Fächerwahl in der Oberstufe über ein breit angelegtes Wissen zu den Grundlagen der Informations- und Kommunikationstechnologien, auf die im Unterricht der Sekundarstufe II aufgebaut werden kann.

Das Fach Informatik wird in der Sekundarstufe II am Couven-Gymnasium als Wahlfach angeboten. In der Einführungsphase können die Schülerinnen und Schüler das Fach als dreistündigen Grundkurs belegen. In der Qualifikationsphase bietet das Gymnasium nun schon mehrere Jahre nacheinander einen neben einem dreistündigen Grundkurs auch einen fünfständigen Leistungskurs im Fach Informatik an. Durch die Kooperation mit den beiden Gym-

nasien *Sankt Leonard* und *Kaiser Karl* sind somit auch immer Schüler dieser Schulen dabei und es besteht eine enge Absprache mit beiden Gymnasien bzgl. des Lehrplans in der EF.

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

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

Darüber hinaus trägt er zu einer breitgefächerten Allgemeinbildung bei, bietet gleichzeitig Raum für individuelle Spezialisierungen und ermöglicht verantwortungsvolles Handeln in einer sich schnell wandelnden und von technischen Fortschritten geprägten Welt. Exkursionen zu speziell für Schüler gemachte Angebote der RWTH im MINT-Bereich (Infosphere, Rechenzentrum, DLR) bieten zudem Möglichkeiten der Berufsorientierung.

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 stellt einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar und hat im Rahmen der Umstellung auf die Programmiersprache Java noch mal eine Belebung erfahren.

Zurzeit besteht die Fachschaft Informatik des Couven-Gymnasiums aus vier Lehrkräften, wobei drei eine Sek-II-Lehrbefähigung besitzen und eine Lehrkraft eine Sek-I-Lehrbefähigung. Das Couven-Gymnasium verfügt über vier Computerräume mit insgesamt 104 Windows-Rechnern (Terminalserverlösung). Zusätzlich stehen 12 MacBooks und 35 TerraPads zur Verfügung, die frei ausgeliehen werden können. Ein in der gesamten Schule verfügbares WLAN ermöglicht somit die Recherche in allen Klassenräumen. 12 iPads dienen zusätzlich dem besonderen Einsatz im Medienbereich, ebenso wie 30 Stereo-Headsets mit Mikrofon. Alle Computerarbeitsplätze sind an ein Intranet angeschlossen, so dass jede Schülerin und jeder Schüler einen individuellen Account besitzt, solange sie/er das Couven-Gymnasium besucht. Die Lernplattform Fronter wird in allen Unterrichtsfächern zum Austausch von Lernmaterialien bzw. zur Abgabe von Ergebnissen eingesetzt. Sie ist auch von Zuhause aus über das Internet erreichbar, ebenfalls über individuelle Accounts.

In den Kursen des Faches Informatik werden zwischen 15 und 25 Schülerinnen und Schüler unterrichtet. Leistungskurse sind in der Regel kleiner. Die Arbeit am Rechner erfolgt also im Normalfall in Einzel- oder Partnerarbeit.

- Fachvorsitzender: Henning Buhr
- Stellvertreter: Ute Hans
- Beauftragter für technische Fragen: Karl Gatzweiler
- Betreuung der Fronter-Plattform: Henning Buhr
- Betreuung der Schulhomepage: Rainer Siemund / Christoph Zock

## 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 Zuordnungen 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 Spielraum 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 nur 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, besitzt die exemplarische Ausweisung "konkretisierter Unterrichtsvorhaben" (Kapitel 2.1.2) empfehlenden Charakter. 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, fachübergreifenden Kooperationen, Lernmitteln und Lernorten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind. Abweichungen von den vorgeschlagenen Vorgehensweisen bezüglich der konkretisierten Unterrichtsvorhaben sind im Rahmen der pädagogischen Freiheit der Lehrkräfte jederzeit möglich. Sicherzustellen bleibt allerdings auch hier, dass im Rahmen der Umsetzung der Unterrichtsvorhaben insgesamt alle konkretisierten Kompetenzerwartungen des Kernlehrplans Berücksichtigung finden.

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

#### 2.1.1 Übersichtsraster Unterrichtsvorhaben

##### 1) Einführungsphase – Grundkurs

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

Zu jedem Unterrichtsvorhaben ist eine Anknüpfung an den Kernlehrplan Informatik in Form von Kompetenzbezügen gegeben. Die aufgeführten Kompetenzen sind dabei so zu verstehen,

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

## Übersichtsraster Unterrichtsvorhaben: Einführungsphase (EF):

<p><b>Unterrichtsvorhaben E-I</b> Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</p> <p>Zentrale Kompetenzen: Argumentieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Einzelrechner Dateisystem Internet Einsatz von Informatiksystemen</p> <p>Zeitbedarf: 2 Stunden</p>	<p><b>Unterrichtsvorhaben E-II</b> Thema: Grundlagen der objektorientierten Analyse und Programmierung anhand von Beispielkontexten</p> <p>Zentrale Kompetenzen: Modellieren Darstellen und Interpretieren Implementieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 14 Stunden</p>
<p><b>Unterrichtsvorhaben E-III</b> Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache Dateisystem (Rechteverwaltung) Internet</p> <p>Zeitbedarf: 12 Stunden</p>	<p><b>Unterrichtsvorhaben E-IV</b> Thema: Modellierung und Implementierung von Spielen mit der Lernumgebung Greenfoot/GLOOP</p> <p>Zentrale Kompetenzen: Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Klassen, Objekte und ihre Beziehungen Analyse, Entwurf und Implementierung Syntax und Semantik einer Programmiersprache API</p> <p>Zeitbedarf: 12 Stunden oder mehr</p>

<p><b>Unterrichtsvorhaben E-V</b> Thema: Implementierung und Entwurf graphischer Benutzeroberflächen zur Vertiefung objektorientierter Programmierung unter Nutzung vorgegebener Schnittstellen (API)</p> <p>Zentrale Kompetenzen: Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p> <p>Zeitbedarf: 10 Stunden</p>	<p><b>Unterrichtsvorhaben E-VI</b> Thema: Verschlüsselung von Daten (Kryptographie)</p> <p>Zentrale Kompetenzen: Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher kryptographischer Verfahren Syntax und Semantik von Programmiersprachen</p> <p>Zeitbedarf: 6 Stunden</p>
<p><b>Unterrichtsvorhaben E-VII</b> Thema: Verwaltung großer Datenmengen – Such- und Sortieralgorithmen</p> <p>Zentrale Kompetenzen: Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Informatiksysteme Formale Sprachen und Automaten</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Algorithmen zum Suchen und Sortieren Dateisystem Einsatz von Informatiksystemen Wirkung der Automatisierung</p> <p>Zeitbedarf: 20 Stunden</p>	<p><b>Unterrichtsvorhaben E-VIII</b> Thema: Geschichte der digitalen Datenverarbeitung</p> <p>Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Formale Sprachen und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Syntax und Semantik einer Programmiersprache Digitalisierung Einzelrechner Wirkung der Automatisierung Geschichte der automatischen Datenverarbeitung</p> <p>Zeitbedarf: 6 Stunden</p>

**Additum:** rekursive Algorithmen

## II) Qualifikationsphase – Grundkurs und Leistungskurs

<b>Qualifikationsphase I</b>	
<p>Unterrichtsvorhaben Q1-I Thema: Rekursive und iterative Algorithmen im Anwendungskontext Sortieren und Suchen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Algorithmen Formale Sprachen und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Grenzen der Automatisierung</p> <p>Zeitbedarf: 15 Stunden</p>	<p>Unterrichtsvorhaben Q1-II Thema: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen</p> <p>Zeitbedarf: 8 Stunden</p>
<p>Unterrichtsvorhaben Q1-III Thema: Modellierung und Implementierung linearer dynamischer Listenstrukturen (Liste, Stapel, Schlange) und deren Anwendungen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme</p> <p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten</p>	<p>Unterrichtsvorhaben Q1-IV Thema: Endliche Automaten und Formale Sprachen</p> <p>Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren</p> <p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprache und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft</p> <p>Inhaltliche Schwerpunkte: Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen</p>

Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen	Sicherheit Wirkung der Automatisierung
Zeitbedarf: 20 Stunden	Zeitbedarf: 20
Unterrichtsvorhaben Q1-V Thema: Maschinennahe Programmierung und Grenzen der Berechenbarkeit	
Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren	
Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprache und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft	
Inhaltliche Schwerpunkte: Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Sicherheit Wirkung der Automatisierung	
Zeitbedarf: 10 Stunden	
<b>Summe Qualifikationsphase I: 73</b>	
<b>Qualifikationsphase II</b>	
Unterrichtsvorhaben Q2-I ( <i>im LK als Q1IIIb integriert</i> ) Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen	Unterrichtsvorhaben Q2-II Thema: Kommunikation, Sicherheit und Datenschutz in Netzstrukturen
Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren	Zentrale Kompetenzen: Argumentieren Darstellen und Interpretieren Kommunizieren und Kooperieren
Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme	Inhaltsfelder: Informatiksysteme Informatik, Mensch und Gesellschaft
Inhaltliche Schwerpunkte: Objekte und Klassen	Inhaltliche Schwerpunkte: Einzelrechner und Rechnernetzwerke Sicherheit Nutzung von Informatiksystemen, Wirkungen der Automatisierung
	Zeitbedarf: 16 Stunden



Analyse, Entwurf und Implementierung von Algorithmen Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen  Zeitbedarf: 20 Stunden	
Unterrichtsvorhaben Q2-III Thema: Kryptographie  Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren  Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprache und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft  Inhaltliche Schwerpunkte: Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Sicherheit Wirkung der Automatisierung  Zeitbedarf: 10 Stunden	Unterrichtsvorhaben Q2-IV Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten  Zentrale Kompetenzen: Argumentieren Modellieren Implementieren Darstellen und Interpretieren Kommunizieren und Kooperieren  Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprache und Automaten Informatiksysteme Informatik, Mensch und Gesellschaft  Inhaltliche Schwerpunkte: Datenbanken Algorithmen in ausgewählten informatischen Kontexten Syntax und Semantik einer Programmiersprache Nutzung von Informatiksystemen Sicherheit Wirkung der Automatisierung  Zeitbedarf: 15 Stunden
<b>Summe Qualifikationsphase II: 61</b>	

### 2.1.2 Konkretisierte Unterrichtsvorhaben

Hinweis: Thema, Inhaltsfelder, inhaltliche Schwerpunkte, Kompetenzen und Absprachen zur vorhabenbezogenen Konkretisierung hat die Fachkonferenz verbindlich vereinbart. In allen anderen Bereichen (Unterrichtssequenzen und verwendeten Beispiele, Medien und Materialien) sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich. Darüber hinaus enthält dieser schulinterne Lehrplan in den Kapiteln 2.2 bis 2.3 übergreifende sowie z. T. auch jahrgangsbezogene Absprachen zur fachmethodischen und fachdidaktischen Arbeit, zur Leistungsbewertung und zur Leistungsrückmeldung. Je nach internem Steuerungsbedarf können solche Absprachen auch vorhabenbezogen vorgenommen werden.

#### I) Einführungsphase – Grundkurs

Im Folgenden sollen die in Abschnitt 1 aufgeführten Unterrichtsvorhaben konkretisiert werden. Diese Konkretisierung hat vorschlagenden Charakter, ohne die pädagogische Freiheit des Lehrenden einschränken zu wollen.

Die übergeordneten Kompetenzen des Kompetenzbereichs "Kommunizieren und Kooperieren" werden in jedem Unterrichtsvorhaben erworben bzw. vertieft und sind daher nicht jedes Mal erneut aufgeführt.

## Kommunizieren und Kooperieren (K)

### Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte,
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit,
- präsentieren Arbeitsabläufe und Arbeitsergebnisse.

Ebenso bieten fast alle Unterrichtsvorhaben, in denen Programme implementiert werden, die Gelegenheit, die folgenden Kompetenzen zu erwerben bzw. zu vertiefen:

### Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellung Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D),
- analysieren und erläutern eine objektorientierte Modellierung (A),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),
- analysieren und erläutern einfache Algorithmen und Programme (A),
- modifizieren einfache Algorithmen und Programme (I),
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- testen Programme schrittweise anhand von Beispielen (I),
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- nutzen das verfügbare Informatiksystem zu strukturierter Verwaltung und gemeinsamer Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K),
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K),
- nutzen die im Unterricht eingesetzten Informatiksysteme selbständig, sicher, zielführend und verantwortungsbewusst (D).

Da in der Einführungsphase das Hauptaugenmerk auf die Einführung der objektorientierten Programmiersprache liegt, werden die oben angegebenen Kompetenzbezüge nicht mehr explizit bei den einzelnen Themenblöcken genannt.

## **Unterrichtsvorhaben E-I**

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Leitfragen: Welche Informatiksysteme werden im Unterricht zwecks Strukturierung des Arbeitsprozesses eingesetzt? Was sind die Grundbausteine der von-Neumann-Architektur und wie greifen sie in einander?

Inhaltliche Schwerpunkte:

Einsatz von Informatiksystemen im Informatik-Unterricht

- Einzelrechner
  - Anmeldevorgang am System
  - EVA-Prinzip, Bedeutung und Auftreten von Informatiksystemen heutzutage
- Dateisystem
  - Innerschulische Strukturen zu Dateiablage und –austausch.
- Internet
  - Umgang mit der Lernplattform *Fronter*

Inhaltsfelder: Inhaltsfeld 3 (Informatik, Mensch und Gesellschaft), Inhaltsfeld 4 (Informatiksysteme)

Zeitbedarf: 2 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Unter Rückgriff auf mittels der Lernplattform zur Verfügung gestellte Materialien stellen die SuS verschiedene Wirkungsfelder und Teilgebiete von Informatiksystemen zusammen und auf Fronter zur Verfügung. Anschließend werden die Ergebnisse vorgestellt, hinterfragt und auf Basis von Bild- und Filmmaterial ergänzt. Anschließend erstellen die SuS eine Definition der Wissenschaft Informatik, die mit existenten Definitionen in Abgleich gebracht wird.

Lernmittel / Materialien:

- Präsentation zu Wirkungsfeldern und Teilgebieten der Informatik
- ABs - Einführung in Fronter
- Filmmaterial der FH-Dortmund und „Informatik-Wer wie was?“
- Definitionen zur Informatik aus der Fachdidaktik und der Gesellschaft für Informatik

<b>Unterrichtssequenzen</b>	<b>zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
1. Einführung in den Umgang mit unserem Rechner-system und der Lernplattform Fronter anhand der Aufgabe, die Bedeutung und die Teilgebiete der Informatik im Rahmen einer Mindmap zusammen zu stellen mit anschließendem Abgleich und Diskussion.	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• kommunizieren die Bedeutung der Informatik für unsere heutige Gesellschaft (K)</li> <li>• Nutzen das Informatiksystem unserer Schule ziel-führend und beschaffen sich nötige Informationen aus dem Internet</li> </ul>	AB „Einführung in Fronter“ Computer  Präsentation
2. Erarbeitung zur Definition der Informatik und Ab-gleich mit den Definitionen der Fachdidaktik	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• kommunizieren die die Informatik konstituieren-den Eigenschaften und Teilaspekte (K)</li> </ul>	
3. Vorstellung der Disziplin Informatik mit anschlie-ßender Diskussion und Meinungs-austausch im Spiegel der Schülerergebnisse	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• kommunizieren ihre Ergebnisse (K)</li> </ul>	Filme

## ***Unterrichtsvorhaben E-II***

Thema: Grundlagen der objektorientierten Analyse und Programmierung anhand von Beispielspielskontexten

Leitfragen: Wie kann ein Problem im informatischen System abgebildet werden? Wie modelliere und implementiere ich in der Programmiersprache Java?

Inhaltliche Schwerpunkte:

- Objekte und Klassen
  - Grundbegriffe objektorientierter Programmierung
  - Unterscheidung zwischen Klasse und Objekt
- Syntax und Semantik einer Programmiersprache / Analyse, Entwurf und Implementierung einfacher Algorithmen
  - Unterscheidung zwischen sondierender und verändernder Methode und Konstruktor
  - Unterscheidung zwischen grundlegenden Datentypen wie Zahlen, Wahrheitswerte und Zeichenketten
  - Anwendung der Kontrollstrukturen Einfachverzweigung und Zählschleife

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 3 (Formale Sprachen und Automaten)

Zeitbedarf: 14 Stunden

Abspraken zur vorhabenbezogene Konkretisierung:

Ausgehend von einem vorgegebenen Projekt zur Erstellung von Zeichnungen aus Quadraten, Kreisen und Dreiecken wird der Unterschied zwischen einer Klasse und einem Objekt dargestellt. Innerhalb dieses Projekts werden eigene Klassen erstellt, indem Quelltexte kopiert und geringfügig abgeändert werden. Objekte werden zur Laufzeit von anderen Objekten erstellt und verändert.

Die verschiedenen Arten von Methoden (sondierende, verändernde Methode, Konstruktor) werden erkannt und verwendet.

Grundlegende Datentypen für Zahlen, Wahrheitswerte und Zeichenketten werden erkannt und im Programm verarbeitet. Der Unterschied zwischen lokalen, globalen Variablen und formalen Parameter wird dargestellt.

Erlerntes wird in verschiedenen vorgegebenen und vorimplementierten Kontexten (Laborkurse, Ticketautomat) reproduziert und in freien Aufgabenstellungen (Heizungssteuerung, Bodymaßindex-Waage, Büchereiprojekt, Spielesammlung mit Schere-Stein-Papier und Zahlenraten) reorganisiert.

Lernmittel / Materialien:

- Barnes/Kölling: Objektorientierte Programmierung mit Java, Pearson-Verlag, 2003
- D. Garmann: Skript "Informatik der Einführungsphase"
- Ulrich Helmich: Informatik 1 mit BlueJ - Ein Kurs für die Stufe 10 / 11 - Teil 1
- Entwicklungsumgebung BlueJ

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Grundbegriffe der objektorientierten Programmierung</p> <p>a) Nutzung von Klassen und Objekten im vorgegebenen Programm</p> <p>b) Analyse von Objekten</p> <p>c) Erarbeitung der Begriffe "Klasse" und "Objekt"</p> <p>d) Erstellung neuer Klassen aus vorhandenen Klassen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen [oder lineare Datensammlungen] zu (M).</li> </ul>	<p>Skript: "Informatik in der Einführungsphase" bzw. Barnes/Kölling Projekte „Laborkurs“, „Figuren“, "Zeichnung"</p> <p>Vertiefung aus Leitprogramm</p>
<p>2. Methoden einer Klasse</p> <p>a) Analyse von vorgegebenen Methoden</p> <p>b) Erstellung eigener Methoden nach vorgegebenen Muster</p> <p>c) Erarbeitung des Begriffs <i>Konstruktor</i></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen [oder lineare Datensammlungen] zu (M),</li> <li>ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>stellen den Zustand eines Objekts dar (D).</li> </ul>	<p>Vertiefung aus Leitprogramm</p>
<p>3. Variablen / Datenfelder einer Klasse und einfache Kontrollstrukturen</p> <p>a) Unterscheidung zwischen globalen Variablen (Datenfeldern) und lokalen Variablen sowie formalen Parametern</p> <p>b) Nutzung von Variablen und Parametern zur Erweiterung des Funktionsumfangs eines vorgegebenen Programms</p> <p>c) Erarbeitung der Begriffe "lokale" und "globale Variable" sowie "formaler Parameter"</p> <p>d) Unterscheidung der Datentypen für Zahlen, Zeichenketten und Wahrheitswerte</p> <p>e) die Kontrollstruktur "Einfachverzweigung"</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen [oder lineare Datensammlungen] zu (M),</li> <li>ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>stellen den Zustand eines Objekts dar (D).</li> </ul>	<p>Projekt "Ticketautomat"</p> <p>Projekte „Bücherei“, „Bodymass-Index“, „Heizungssteuerung“</p> <p>Projekte zur Implementation geometrischer Fragestellungen</p> <p>Vertiefung aus Leitprogramm</p>
<p>4. Vertiefung der Grundlagen zur objektorientierten Programmierung</p> <p>a) Erstellung neuer Klassen ohne Vorgabe</p> <p>b) die Kontrollstruktur "bedingte Wiederholung"</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <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> </ul>	<p>Projekte „ggT“, „kgV“ u.a. (siehe Skript)</p> <p>Projekt „Spielesammlung“</p> <p>Vertiefung aus Leitprogramm</p>

## ***Unterrichtsvorhaben E-III***

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen sowie einfacher Kontrollstrukturen

Leitfragen: Wie lassen sich komplexere Gegebenheiten durch mehrere Klassen darstellen und wie stehen diese in Beziehung zueinander?

Inhaltliche Schwerpunkte:

- Objekte und Klassen:
  - Grundbegriffe objektorientierter Programmierung
  - Vererbung von Klassen
  - Objektinteraktion durch Assoziation
- Analyse, Entwurf und Implementierung einfacher Algorithmen / Syntax und Semantik einer Programmiersprache
  - Anwendung der Kontrollstrukturen Einfachverzweigung und Schleife mit Anfangsbedingung
  - Referenzvariablen

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

Zeitbedarf: 12 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Anhand einfacher Beispiele aus der Erfahrungswelt der Schülerinnen und Schüler wird der Sinn verschiedener Klassen thematisiert und ihre verschiedenen Beziehungen untereinander. Hier wird vor allem Augenmerk auf die Vererbung gelegt und die damit einhergehende Kapselung der Daten über Modifikatoren. Die Möglichkeit der Assoziativen Beziehung zwischen Klassen ist unausweichlich mit zu behandeln, spätestens aber im folgenden Unterrichtsprojekt.

Lernmittel / Materialien:

- Verschiedene Beispiele aus „Informatik 2“, Schöningh
- Verschiedene Beispiele aus „Java von Kopf bis Fuß“, O'Reilly
- Ulrich Helmich: Informatik 1 mit BlueJ - Ein Kurs für die Stufe 10 / 11 - Teil
- Entwicklungsumgebung BlueJ

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Grundlagen der Vererbung und ihre Darstellung in UML als Entwurfs- bzw. Implementationsdiagramm sowie Implementation derselben	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• modellieren Klassen mit UML mit ihren Attributen und Methoden(M),</li> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen [oder lineare Datensammlungen] zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich (Modifikatoren) zu (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• stellen den Zustand eines Objekts dar (D)</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>• implementieren kleinere Programme und testen die Vererbungs- und Assoziationsbeziehungen.</li> </ul>	<p>Beispiele aus den Büchern:</p> <p>- Informatik 2, Schöningh</p> <p>-Java von Kopf bis Fuß, O'Reilly</p>
2. Erweiterung der Klassenbeziehungen durch Assoziation und ihre Darstellung als UML sowie Implementation derselben		



## Unterrichtsvorhaben E-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen sowie einfacher Kontrollstrukturen

Leitfragen: Wie lassen sich Spiele unter Berücksichtigung von Tastatureingaben realisieren? Wie werden hier Klassen und ihre Objekte verwaltet?

Inhaltliche Schwerpunkte:

- Objekte und Klassen:
  - Grundbegriffe objektorientierter Programmierung
  - Vererbung von Klassen
  - Objektinteraktion
- Analyse, Entwurf und Implementierung einfacher Algorithmen / Syntax und Semantik einer Programmiersprache
  - Anwendung der verschiedenen Kontrollstrukturen im komplexen Kontext
  - Referenzvariablen
- Dateisystem / Internet:
  - Verwendung von Klassendokumentationen und Hilfestellungen aus dem Internet

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

Zeitbedarf: 12 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Ausgehend von einem vorgegebenen Greenfootprojekt (z. B. little crab oder crazy kara) wird die Entwicklungsumgebung Greenfoot eingeführt. Innerhalb des Projekts werden die Klassen schrittweise durch Vererbung konkretisiert und deren Funktion implementiert. Dabei werden die bereits erlernten Grunddatentypen und verschiedenen Arten von Methoden in anderem Zusammenhang angewendet und eingeübt. Insbesondere wird die Kapselung sowie der Zugriff auf Klassenmethoden mittels Objekten vertieft.

Erlernes wird in einem komplett selbst erstellten Spiel reorganisiert. Das Projekt wird dabei geeignet dokumentiert.

Lernmittel / Materialien:

- Entwicklungsumgebung Greenfoot
- Material „little crab“: <http://www.ralphhenne.de/informatik/greenfoot/01Krabbe.pdf>
- Greenfoot Simulation: <http://www.ralphhenne.de/informatik/greenfoot/05HaseFuchs.pdf>
- Allgemein: <http://www.ralphhenne.de/informatik/greenfoot/greenfoot.html>
- Night Hacking with Michael Kölling: <http://www.youtube.com/watch?v=HbCLs44PcZE>
- Greenfoot-Video-Tutorials von Michael Kölling: <http://blogs.kent.ac.uk/mik/category/joy-of-code/>
- Greenfoot Video-Tutorials auf Deutsch: <http://www.greenfoot-center.de/index.php/news/81-greenfoot-videotutorials-auf-deutsch>

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Grundlagen der Entwicklungsumgebung Greenfoot</p> <p>a) Darstellung / Verhalten von Objekten (act-Methode)</p> <p>b) Grundlegende Methoden der Klassen Actor, Animal und World</p> <p>c) Objektinteraktion</p>	<p>Die Schülerinnen und Schüler</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>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen [oder lineare Datensammlungen] zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• stellen den Zustand eines Objekts dar (D),</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I).</li> </ul>	<p>Ausgangsprojekt: z. B. "little crab" oder "crazy kara"</p> <p>Das Projekt "little crab" ist in der Installation des Greenfoot-Paketes enthalten. Das Projekt "crazy kara" kann von Schulhomepage heruntergeladen werden.</p> <p>Konkrete Umsetzungsvorschläge gibt es zudem unter:  <a href="http://www.ralphhenne.de/informatik/greenfoot/01Krabbe.pdf">http://www.ralphhenne.de/informatik/greenfoot/01Krabbe.pdf</a></p>
<p>2. Weitere Möglichkeiten von Greenfoot</p> <p>a) Tastatursteuerung</p> <p>b) Sounds</p> <p>c) bewegte Grafiken</p> <p>d) Counter</p> <p>e) Objekte erstellen neue Objekte</p>	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K),</li> <li>• testen Programme schrittweise (I),</li> <li>• implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	<p>Hier könnte sehr motivierten Schülern die Homepage von Michael Kölling (joy of code) mit den Lernvideos empfohlen werden, wo sie Ideen für ihr eigenes Spiel entwickeln können.</p>
<p>3. Erstellung einer eigenen Animation/eines eigenen Spiels</p> <p>a) Entwicklung einer Spiel-/Animationsidee (Anforderungsdefinition)</p> <p>b) Implementierung der Idee unter Verwendung von Hilfestellungen aus dem Internet</p> <p>c) Dokumentation des Projekts</p>	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K),</li> <li>• testen Programme schrittweise (I),</li> <li>• implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	<p>Schülern soll hier freier Raum zur Entfaltung gegeben werden. Anregungen können sie sich auf der Homepage von Michal Kölling – joy of code – holen.</p> <p>Weitere Anregungen: Homepage von Ralph Henne: Simulation zweier Populationen, die sich vermehren und in einer Räuber-Beute-Abhängigkeit koexistieren. Material/Idee siehe:  <a href="http://www.ralphhenne.de/informatik/greenfoot/05HaseFuchs.pdf">http://www.ralphhenne.de/informatik/greenfoot/05HaseFuchs.pdf</a></p>

## ***Unterrichtsvorhaben E-V***

Thema: Implementierung und Entwurf graphischer Benutzeroberflächen zur Vertiefung objektorientierter Programmierung unter Nutzung vorgegebener Schnittstellen (API)

Leitfragen: Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren? Was beinhalten die Abläufe und Inhalte der Anwendung aus fachlicher Sicht? Wie hängen die Aktivitäten und ihre Inhalte strukturell in der Anwendung zusammen?

Inhaltliche Schwerpunkte:

- Objekte und Klassen:
  - GUI-Design und Ereignisbehandlung -> MVC-Konzept
- Analyse, Entwurf und Implementierung einfacher Algorithmen / Syntax und Semantik einer Programmiersprache:
  - Anwendung der Kontrollstrukturen Schleife mit Anfangsbedingung und mit Endbedingung
  - Analyse, Modifikation und Implementation mathematischer Algorithmen
  - Darstellung des Ablaufs von Algorithmen

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten)

Zeitbedarf: 20 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Mithilfe des JavaEditors (PC), Eclipse bzw. BlueJ werden grafische Oberflächen erstellt und mit den in den Unterrichtsvorhaben E-II erstellten Algorithmen verknüpft. Dabei werden die Ereignisbehandlungsroutinen der Oberflächen-Objekte verwendet und das Geheimnisprinzip angewendet. Das Model-View-Controller-Konzept findet Anwendung.

Lernmittel / Materialien:

- D. Garmann: Skript "Informatik der Einführungsphase"
- Entwicklungsumgebung BlueJ
- <http://www.ralphhenne.de/informatik/bluej/bluej.html>
- Standardleitfragen für GUI: <http://www.eosp.de/papers/user-interface-architektur.pk2.pdf>

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Erstellung von Anwendungen</p> <ul style="list-style-type: none"> <li>• Ausgehend vom Entwurfsdiagramm über das Implementationsdiagramm wird ein Programm zum Spieleautomaten entworfen</li> <li>• GUI-Entwurf <ul style="list-style-type: none"> <li>○ Komponenten (Buttons, Textfelder, ...)</li> <li>○ Layout-Manager (Grid-Layout, BorderLayout, ...)</li> <li>○ ActionListener</li> </ul> </li> <li>• Model-View-Controller-Konzept <ul style="list-style-type: none"> <li>○ Gegebenenfalls auch in b) integriert.</li> </ul> </li> </ul>	<p>Die Schülerinnen und Schüler</p> <ol style="list-style-type: none"> <li>a) modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>b) ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>c) stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>d) stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>e) implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>f) modifizieren und erweitern Computerprogramme (I),</li> <li>g) testen und korrigieren Computerprogramme (I).</li> </ol>	<p>Projekt: Spieleautomat (Skript 11.1 – Java Garmann)</p> <p>Die Schüler erstellen eine Anwendung, welche die zuvor erlernten Techniken der Vererbung und Assoziation von Klassen umsetzt. Durch die Einführung in die GUI werden die Schülerinnen und Schüler befähigt ansprechende Anwendungen zu programmieren. Die Grafische Oberfläche selbst besitzt dabei keine "eigene Funktionalität", außer der Aufgabe, entsprechende Methoden der zuvor entwickelten Klasse aufzurufen.</p>

## ***Unterrichtsvorhaben E-VI***

Thema: Verschlüsselung von Daten (Kryptographie)

Leitfragen: Wie können große Daten geheim übermittelt werden? Welchen Antrieb hatten / haben Menschen, ihre Daten geheim übermitteln zu wollen? Seit wann gibt es Kryptographie?

Inhaltliche Schwerpunkte:

- Objekte und Klassen:
  - lineare Datensammlung von Objekten
- Dateisystem:
  - Funktionsweise von Textdateien
- Algorithmen zum Ver- und Entschlüsseln von Nachrichten:
  - Caesar und Vigenère
- Einsatz von Informatiksystemen / Wirkung der Automatisierung:
  - Datenschutz und Sicherheit

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

Zeitbedarf: 6 Stunden

Abspraken zur vorhabenbezogene Konkretisierung:

Am Beispiel des Kampfes zwischen Römern und Galliern wird auf das damalige Bedürfnis der Römer, ihre Nachrichten geheim zu übermitteln, eingegangen. Der Caesar-Code und seine Arbeitsweise wird thematisiert. In diesem Zusammenhang werden mit den SuS benötigte String-Verarbeitungsmethoden erarbeitet. Sie entwickeln einen Algorithmus zum Verschlüsseln von Klartext und zum Entschlüsseln von Geheimtext und implementieren ihn. Dabei kommt auch der Modul-Operator zum Einsatz. Caesar wird analysiert bzgl. seiner Sicherheit und weiterentwickelt zum Vigenère-Code. Schnelle SuS implementieren diesen ebenfalls. Die Implementierung erfolgt in beiden Fällen objektorientiert.

Die Auswirkungen des Einsatzes von Informatiksystemen (Algorithmen zur Codierung / Verschlüsselung) werden thematisiert. Dies kann z.B. anhand der IBAN oder EAN mit Bezug auf unsere Gesellschaft geschehen und seine Auswirkung auf Arbeitsstrukturen.

Lernmittel / Materialien:

- Diverse Lehrbücher der Informatik, wie Klett oder Schöningh
- Film: Verschlüsselungsverfahren nach Caesar und Vigenere (<http://www.youtube.com/watch?v=GkgxbvqqrNQ>)
- Film: Mathematik zum Anfassen : Kryptographie – Prof. Albrecht Beutelspacher (<http://www.youtube.com/watch?v=VeH0KnZtljY>)  
Entwicklungsumgebung BlueJ bzw. Eclipse (mit GUI)

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Problemanalyse Caesar-Verschlüsselung</p> <ul style="list-style-type: none"> <li>• Modularisierung des Caesar-Codes</li> <li>• Modellierung des Caesar-Codes</li> <li>• Entwicklung einer GUI</li> <li>• Verknüpfung der GUI mit dem Modell</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ol style="list-style-type: none"> <li>a) modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>b) ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>c) ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>d) stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>e) stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>f) dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).</li> </ol>	<p>Einstieg in die Thematik mit einem Film von Prof. A. Beutelspacher zur Kryptographie oder einem Comicausschnitt aus Asterix (Fronter)</p> <p>Diverse Informatikbücher als Anhaltspunkt.</p>
<p>2. Algorithmen zum Ver- und Entschlüsseln</p> <ul style="list-style-type: none"> <li>• Lesen von Textdateien (Buchstabe für Buchstabe)</li> <li>• Nutzung von ASCII-Zeichen zur Umwandlung</li> <li>• Modulo-Operator</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ol style="list-style-type: none"> <li>a) dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D),</li> <li>b) analysieren den Nutzen von ASCII-Tabellen und wenden sie auf das Beispiel an (D).</li> </ol>	<p>Diverse Informatikbücher als Anhaltspunkt. Caesar-Verschlüsselung- ein kleines Tutorial (<a href="http://www.christophtutorials.de/beitrag.php?beitrag=Java_-_Caesar-Verschlueselung">http://www.christophtutorials.de/beitrag.php?beitrag=Java_-_Caesar-Verschlueselung</a>)</p>
<p>3. Textdateien</p> <ul style="list-style-type: none"> <li>• String-Verarbeitung</li> <li>• Bedeutung von ASCII-Tabellen</li> <li>• Auswirkungen des Einsatzes von Informatiksystemen</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D),</li> <li>• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).</li> </ul>	<p>Fallbeispiele zum Thema Codes (EAN, ISBN) und ihre Auswirkung auf die Arbeitswelt. Bildungsserver Informatik Rheinland-Pfalz (<a href="http://informatik.bildung-rp.de/fileadmin/user_upload/informatik.bildung-rp.de/Fortbildung/pdf/NET-111114-Schlemmer-AB-Inf11-Pruefziffern.pdf">http://informatik.bildung-rp.de/fileadmin/user_upload/informatik.bildung-rp.de/Fortbildung/pdf/NET-111114-Schlemmer-AB-Inf11-Pruefziffern.pdf</a>)</p> <p>Klett „Informatik 5“ - EAN-Beispiel könnte wieder aufgegriffen werden bei der Automaten-theorie! (<a href="http://www2.klett.de/sixcms/media.php/71/873277/731068">http://www2.klett.de/sixcms/media.php/71/873277/731068</a> Informatik 5 KI netz.pdf)</p>

## **Unterrichtsvorhaben E-VII**

Thema: : Verwaltung großer Datenmengen – Such- und Sortieralgorithmen

Leitfragen: Wie können gleichartige Daten einfach verwaltet werden? Wie können diese Daten effizient analysiert und sortiert werden?

Inhaltliche Schwerpunkte:

- Objekte und Klassen:
  - Vertiefung MVC-Konzept
- Algorithmen zum Sortieren :
  - Analyse von Algorithmen quadratischer Laufzeit
  - Entwurf eigener Sortierverfahren
  - Sortieralgorithmus (Selectionsort: Minsort oder Maxsort)
  - Effizienzbetrachtungen und -vergleiche
- Analyse, Entwurf und Implementierung einfacher Algorithmen:
  - Anwendung der bekannten Schleifen-Kontrollstrukturen und der Mehrfachverzweigung
  - Die Datenstruktur Array im Sachzusammenhang

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen)

Zeitbedarf: 20 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Die einzelnen Bausteine einer Lottosimulation werden Schritt für Schritt analysiert, algorithmisiert und implementiert. Der Datentyp eines eindimensionalen Arrays wird zur Verwaltung gleichartiger Daten eingeführt und verwendet. Die Bausteine der Lottosimulation sind im Einzelnen: Prüfung eines Lottoscheins auf Richtigkeit (keine doppelten Zahlen), Sortierung der Tippzahlen, Ziehung der Lottozahlen, Gewinnrangermittlung.

Das zuvor erlernte MVC-Konzept wird hier angewendet, d. h. es wird strikt zwischen Funktionalität (Controller), Daten (Model) und Oberfläche (View) getrennt.

Zuvor erlernte Kontrollstrukturen werden vertieft und um eine Mehrfachverzweigung ergänzt.

Lernmittel / Materialien:

- D. Garmann: Skript "Informatik der Einführungsphase" (Skript 11-1-Java.pdf, S. 32 - 41)
- Animation *Omozon.jar* – Animation zur leichten Entdeckung von Sortieralgorithmen und deren Umsetzung in Pseudocode (Fronter EF)
- Th. Kempe, A. Löhr: „Informatik 2“, Schöningh, 2012 (S. 114 – 115)
- Entwicklungsumgebung BlueJ

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Problemanalyse Lottosimulation</p> <ul style="list-style-type: none"> <li>• Modularisierung und Modellierung der Lottosimulation, MVC-Konzept</li> <li>• Die Datenstruktur des Array: Model der Lottosimulation</li> <li>• Entwicklung eines GUI der Lottosimulation: View der Lottosimulation</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ol style="list-style-type: none"> <li>a) modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>b) ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>c) ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>d) stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>e) stellen den Zustand eines Objekts dar (D),</li> <li>f) stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>g) dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).</li> </ol>	<p>Skript: "Informatik in der Einführungsphase – Java -Teil 1 S. 32-41" im Fronter-Ordner. Dieses Projekt ist bezüglich des MVC-Konzepts eines der umfangreichsten Projekte, welches die Schülerinnen und Schüler komplett selber erstellen. (Schwachen Gruppen könnte man ein Gerüst zur Verfügung stellen.)</p>
<p>2. Algorithmisierung der Lottosimulation: Controller</p> <ul style="list-style-type: none"> <li>• Prüfung der Korrektheit des Tippscheins</li> <li>• Sortierung der Tippzahlen</li> <li>• Ziehung der Lottozahlen / Quicktipp</li> <li>• Gewinnrangermittlung</li> </ul>	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ol style="list-style-type: none"> <li>a) Entwickeln Sortieralgorithmen und wenden sie auf Beispiele an (D),</li> <li>b) entwerfen zunächst als Pseudocode einen Algorithmus zum Sortieren, evtl. mit Unterstützung einer Animation (<i>Omozon.jar</i>) (M),</li> </ol>	<p>Skript: "Informatik in der Einführungsphase – Java -Teil 1 S. 32-41" im Fronter-Ordner. Evtl. Animation zur Unterstützung des Prozesses: <i>Omozon.jar</i> im Fronter-Ordner.</p>
<p>3. Abschluss Lottosimulation</p> <ol style="list-style-type: none"> <li>a) Verknüpfung zw. View, Model und Controller</li> <li>b) Verbesserung der grafischen Oberfläche durch komfortable Gewinnrangausgabe oder Tippscheinabgabe</li> <li>c) Nutzung anderer Sortierverfahren</li> </ol>	<ol style="list-style-type: none"> <li>c) beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A) – dies im Rahmen der für die EF vorgesehenen Weise.</li> </ol>	



## ***Unterrichtsvorhaben E-VIII***

Thema: Geschichte der digitalen Datenverarbeitung

Leitfragen: Welche Entwicklung durchlief die moderne Datenverarbeitung? Wie funktioniert ein moderner Computer im Gegensatz zu früheren Computern?

Inhaltliche Schwerpunkte:

- Syntax und Semantik einer Programmiersprache:
  - Überführung von Java-Quelltexten über Assemblercode zum Maschinencode
- Digitalisierung:
  - Binärdarstellung von ganzen Zahlen
- Einzelrechner
  - Von-Neumann-Architektur
- Geschichte der automatischen Datenverarbeitung
  - geschichtliche Grundlagen der Rechnerentwicklung

Inhaltsfelder: Inhaltsfeld 3 (Formale Sprache und Automaten), Inhaltsfeld 4 (Informatikssysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

Zeitbedarf: 6 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Ausgehend von einem Java-Programm werden schrittweise Ideen entwickelt, wie komplexere arithmetische Ausdrücke und Kontrollstrukturen auf Maschinenebene heruntergebrochen werden können, dabei wird der Focus auf die Möglichkeit der Automatisierung gelegt. Mithilfe des Modellassemblers DC werden die Binärdarstellung von Daten und die Von-Neumann-Architektur veranschaulicht.

Die Von-Neumann-Architektur wird in den geschichtlichen Kontext gesetzt.

Lernmittel / Materialien:

- D. Garmann: Skript "Von der Hochsprache zur Maschinenebene"
- Gierhard: Materialien zum DC unter <http://www.oberstufeninformatik.de/dc/>
- Arbeitsblätter
- Modellassembler DC

<b>Unterrichtssequenzen</b>	<b>zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
1. Von der Hochsprache zum Assembler <ul style="list-style-type: none"> <li>• arithmetische Zuweisungen</li> <li>• Einfachverzweigung</li> <li>• Schleifen mit Anfangsbedingung</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	Skript: " Von der Hochsprache zur Maschinenebene "
2. Vom Assembler zur Maschinenebene <ul style="list-style-type: none"> <li>• Darstellung von Zahlen im Binärsystem</li> <li>• Darstellung von Assemblerbefehlen</li> <li>• Von-Neumann-Architektur</li> </ul>	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li> <li>• beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).</li> </ul>	Skript: " Von der Hochsprache zur Maschinenebene "
3. Geschichte der Datenverarbeitung	Die Schülerinnen und Schüler <ul style="list-style-type: none"> <li>• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).</li> </ul>	Arbeitsblatt zur Geschichte der Datenverarbeitung

## II) Qualifikationsphase - Grundkurs

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

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung  
*/Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen* (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

Ebenso bieten fast alle Unterrichtsvorhaben, in denen Programme implementiert werden, die Gelegenheit, die folgenden Kompetenzen zu erwerben bzw. zu vertiefen:

Schülerinnen und Schüler

- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- wenden eine didaktisch orientierte Entwicklungsumgebung (*didaktisch orientierte Entwicklungsumgebungen*) zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),
- *entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem* (M).

## **Unterrichtsvorhaben Q1-I**

Die rot gedruckten (möglichen) Ergänzungen beziehen sich auf den Leistungskurs.

Thema: Rekursive und iterative Algorithmen im Anwendungskontext Sortieren und Suchen

Leitfragen: Wie können komplexe, rekursiv definierte Probleme informatisch gelöst werden?  
Gibt es schnelle (rekursiv definierte) Sortier- und Suchverfahren?

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
  - Rekursion als neues Instrument für Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
  - Gegenüberstellung bekannter Such- und Sortierverfahren mit iterativer/rekursiver Definition
  - schnelle Sortierverfahren mit Laufzeit  $O(n \cdot \log(n))$
- Syntax und Semantik einer Programmiersprache
  - Rekursive Programmierung
- Nutzung von Informatiksystemen
- Grenzen der Automatisierung
  - Grenzen rekursiver Algorithmen

Inhaltsfelder: Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

Zeitbedarf: 15/20 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Ausgehend vom einem Problem wie z. B. "Türme von Hanoi" wird Rekursion als fundamentale Idee der Informatik zunächst in mathematischem, danach aber auch im informatischen Zusammenhang angewendet. Dabei wird zwischen linearen und verzweigten Rekursionen unterschieden und das Laufzeitverhalten bei hoher Rekursionstiefe analysiert.

Verschiedene NP-vollständige Probleme (wie z. B. Rucksack, n-Damen, Springer, Irrgarten, etc.) werden algorithmisch rekursiv formuliert **und als Backtracking-Algorithmus implementiert**.

Bereits bekannte Such- und Sortierverfahren (z. B. Sortieren durch Einfügen, Sortieren durch Auswahl, Sequentielle Suche) werden rekursiv formuliert und durch leistungsfähigere Verfahren (z. B. Quicksort, Mergesort, **Heapsort**, Binäre Suche) ergänzt. **Die neuen Verfahren werden implementiert**.

Lernmittel / Materialien:

- Arbeitsblätter auf Fronter/Intranet
- eine didaktische Entwicklungsumgebung (z. B. Java-Editor, BlueJ)

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Entwicklung der Rekursion als fundamentale Idee der Informatik</p> <ul style="list-style-type: none"> <li>• Rekursion in mathematischen Kontexten</li> <li>• rekursive Formeln</li> <li>• rekursive Funktionen / Methoden</li> <li>• rekursive Programmierung</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> </ul>	<p>Türme von Hanoi mit Schwerpunkt auf</p> <ul style="list-style-type: none"> <li>• Zahl der Versetzungsoperationen</li> <li>• Protokollierung der Versetzungen</li> </ul>
<p>2. Rekursion in mathematischen und informatischen Kontexten</p> <ul style="list-style-type: none"> <li>• Analyse und Darstellung des rekursiven Ablaufs einer Methode</li> <li>• Analyse des Laufzeitverhaltens linearer und verzweigter Rekursion</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>• testen Programme systematisch anhand von Beispielen <b>und mit Hilfe von Testanwendungen</b> (I).</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<ul style="list-style-type: none"> <li>• Fakultätsfunktion (lineare Rekursion)</li> <li>• Fibonacci-Funktion (verzweigte Rekursion)</li> <li>• ggT (verzweigte Rekursion)</li> <li>• evt. Fraktale (Kochkurve, Sierpinski dreieck, etc.)</li> </ul>
<p>3. NP-vollständige Probleme lösen mit Backtracking</p> <ul style="list-style-type: none"> <li>• Erarbeitung verschiedener NP-vollständiger Probleme</li> <li>• Algorithmische Beschreibung einer Lösungsidee</li> <li>• <b>Implementierung eines Problems als Backtrackingalgorithmus</b></li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ <b>und "Backtracking"</b> (M)</li> <li>• testen Programme systematisch anhand von Beispielen <b>und mit Hilfe von Testanwendungen</b> (I).</li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter und Rasterprogramm zum N-Damenproblem</li> <li>• Arbeitsblätter zum Springerproblem</li> <li>• Rucksackproblem</li> <li>• <b>Projektarbeit zur Irrgartenproblematik</b></li> <li>• Material Buch „Informatik macchiato“ S.75 Wenn es klappt, dann kann es dauern“</li> </ul>
<p>4. Effiziente Sortierverfahren / Suchverfahren</p> <ul style="list-style-type: none"> <li>• Wiederholung bereits bekannter Sortier- und Suchverfahren als rekursiver Algorithmus</li> <li>• Erarbeitung eines Sortierverfahrens der Laufzeit</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrationsprogramm zur Visualisierung von Sortierverfahren z.B. youtube Video „Bubblesort with hungarian</li> </ul>

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

## **Unterrichtsvorhaben Q1-II**

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Leitfragen: *Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen? Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen? Welche objektiven Qualitätsmerkmale gute Modellierungen lassen sich finden?*

### **Vorhabenbezogenen Konkretisierung:**

Zu einer Problemstellung in einem Anwendungskontext soll eine Java-Anwendung entwickelt werden. Die Problemstellung soll so gewählt sein, dass für diese Anwendung die Verwendung einer abstrakten Oberklasse als Generalisierung verschiedener Unterklassen sinnvoll erscheint und eine Klasse durch eine Unterklasse spezialisiert werden kann. Um die Aufgabe einzugrenzen, können (nach der ersten Problemanalyse) einige Teile (Modellierungen oder Teile von Java-Klassen) vorgegeben werden.

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm (**bzw. Standard-UML mit entsprechender Unterscheidung von Komposition, Aggregation etc.**) dargestellt. Dabei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert (**im LK verpflichtend**). Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt. Betont wird die Auseinandersetzung mit den Vor- bzw. Nachteilen verschiedener Modellierungen und der Vorzug der Komposition vor der Vererbung unterstrichen. **Im LK werden gängige Entwurfsmuster im Anwendungskontext erarbeitet und bewertet.**

**Zeitbedarf:** 8 Stunden / **12 Stunden**

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines/<b>mehrerer</b> kontextbezogenen(r) Beispiels/e</p> <p>(a) Analyse der Problemstellung</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm <b>bzw. Standard UML</b>)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p> <p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(g) <b>Modellieren mit Hilfe einfacher Entwurfsmuster</b></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• wägen Vor- und Nachteile verschiedener Modellierungen ab (Komposition vor Vererbung) (M)</li> <li>• <b>setzen Modellierungen mit Hilfe von Entwurfsmustern um (M)</b></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> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung</li> </ul>	<p><i>Beispiel:</i> Wetthuepfen</p> <p>Für ein Wetthüpfen zwischen einem Hasen, einem Hund und einem Vogel werden die Tiere gezeichnet. Alle Tiere springen wiederholt nach links. Die Höhe und Weite jedes Hüpfers ist zufällig. Evtl. marschieren sie anschließend hintereinander her.</p> <p>oder</p> <p><i>Beispiel:</i> Tannenbaum</p> <p>Ein Tannenbaum soll mit verschiedenen Arten von Schmuckstücken versehen werden, die durch unterschiedliche geometrische Objekte dargestellt werden. Es gibt Kugeln, Päckchen in der Form von Würfeln und Zuckerringe in Form von Toren.</p> <p>Ein Prototyp, der bereits mit Kugeln geschmückt werden kann, kann zur Verfügung gestellt werden. Da alle Schmuckstücke über die Funktion des Auf- und Abschmückens</p>



	<p>dokumentierter Klassenbibliotheken (I),</p> <ul style="list-style-type: none"> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>• 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>verfügen sollen, liegt es nahe, dass entsprechende Methoden in einer gemeinsamen Oberklasse realisiert werden.</p> <p><i>Beispiel:</i> Pizza-Belag (Kompositionsansatz)</p> <p><i>Beispiel:</i> Wetterstation (Observer-Pattern)</p> <p><i>Beispiel:</i> Cafe-Shop StarBuzz (Decorator-Pattern)</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.1-Wiederholung</p> <p><a href="#">(Download Q1-I.1)</a></p> <p><i>Entwurfsmuster aus der Reihe Heads First (O'Reilly)</i></p>
--	---	---

## ***Unterrichtsvorhaben Q1-III***

Thema: Modellierung und Implementierung dynamischer Listenstrukturen (Liste, Stapel, Schlange) und deren Anwendungen

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
  - Darstellung der ZA-Klassen im Entwurfs- und Implementationsdiagramm
  - Verwendung der Dokumentation der ZA-Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
  - Verwendung der Standardoperationen der linearen Datenstrukturen
- Algorithmen in ausgewählten informatischen Kontexten
  - Anwendungsbeispiele für Schlange, Stapel, Liste
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen

Inhaltsfelder: Inhaltsfeld 1 (Daten und ihre Strukturierung), Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 3 (Formale Sprachen und Automaten), Inhaltsfeld 4 (Informatiksysteme)

Zeitbedarf: 20/25 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Schlangen am Beispiel dargestellt und die Operationen der Klasse `Queue` erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert.

Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse `List` eingeführt und in einem Anwendungskontext verwendet.

In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Lernmittel / Materialien:

- Arbeitsblätter (auch digital), Buch „Informatik 2 – Modellierung, Datenstrukturen und Algorithmen“, Schöningh
- eine didaktische Entwicklungsumgebung (z. B. Java-Editor, BlueJ, **evtl. Eclipse**)
- Animationen zur Veranschaulichung der ADT's (z.B. Stack:  
[https://www.youtube.com/watch?v=V2Tb\\_v6vuqg](https://www.youtube.com/watch?v=V2Tb_v6vuqg), List:  
[http://www.csanimated.com/animation.php?t=Linked\\_list](http://www.csanimated.com/animation.php?t=Linked_list)

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse <code>Stack</code></p> <p>2. Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Funktionalität der Klasse <code>Stack</code></li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung der Klasse <code>Stack</code>.</li> <li>• <b>Implementierung der Klasse <code>Stack</code></b></li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen (D),</li> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• stellen Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>• implementieren Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• testen Programme systematisch anhand von Beispielen <b>und mit Hilfe von Testanwendungen</b></li> </ul>	<p><i>Anwendung im Alltag: Bearbeiten-Rückgängig</i></p> <p>Beispiele zur Implementierung:</p> <ul style="list-style-type: none"> <li>- Klammerung in modernen Programmiersprachen</li> <li>- Palindrome</li> <li>- Frachtschiffbeladung mit Containern</li> </ul>
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></p> <p>3. Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Funktionalität der Klasse <code>Queue</code></li> <li>• Modellierung und Implementierung der Anwendung unter Verwendung der Klasse <code>Queue</code>.</li> <li>• <b>Implementierung der Klasse <code>Queue</code></b></li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen (D),</li> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• stellen Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>• implementieren Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• testen Programme systematisch anhand von Beispielen <b>und mit Hilfe von Testanwendungen</b></li> </ul>	<p>Beispiele zur Implementierung: Warteschlangen in einer Praxis, Druckerschlange, Verkehrskontrolle</p>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse <code>List</code></p> <p>(h) Erarbeitung der Vorteile der Klasse <code>List</code> im Gegensatz zu den bereits bekannten linearen Strukturen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>• modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>• stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen (D),</li> <li>• analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> <li>• analysieren und erläutern Algorithmen und Programme (A),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• stellen Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>• implementieren Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• testen Programme systematisch anhand von Beispielen <b>und mit Hilfe von Testanwendungen</b></li> </ul>	<p>Beispiele zur Implementierung: Vokabeltrainer, Bowlingbahn</p> <p><b>Material:</b></p> <p>Abiturklassen, Buch Informatik 2,</p>

<p>(i) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse <code>List</code>.</p>	<p>(I).</p> <ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer Datenstrukturen (A),</li> <li>• <b>implementieren ausgewählte Operationen dynamischer Datenstrukturen (I),</b></li> </ul>	<p>Schöningh, weiteres Onlinematerial</p>
<p>4. Vertiefung / Anwendung einer linearen Datenstruktur im Anwendungskontext.</p>	<p>zusätzlich: Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M),</li> </ul>	<p>z.B. Soundplayer mit GUI (siehe Java mit BlueJ von Kölling, Barnes, S. 443ff)</p>

## **Unterrichtsvorhaben Q1-IV**

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?*

### **Vorhabenbezogene Konkretisierung:**

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.

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.

Auch andere Grammatiken werden untersucht, entwickelt oder modifiziert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet.

**Zeitbedarf:** 20 Stunden

## Sequenzierung des Unterrichtsvorhabens:

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 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> <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> </ul>	<p><i>Beispiele:</i></p> <p>Cola-Automat, Geldspielautomat, Roboter, Zustandsänderung eines Objekts „Auto“, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen (<a href="#">Download Q2-II.1</a>)</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 Ent-</p>	<p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Ent-</p>	<p><i>Beispiele:</i></p> <p>reguläre Grammatik für Wörter mit ungerader Parität, Grammatik für Wörter, die bestimmte Zahlen</p>

<p>wicklung 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>	<ul style="list-style-type: none"> <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> </ul>	<p>repräsentieren, Satzgliederungsgrammatik</p> <p><i>Materialien: (s.o.)</i></p>
<p><b>3. Grenzen endlicher Automaten</b></p>	<ul style="list-style-type: none"> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).</li> </ul>	<p><i>Beispiele:</i></p> <p>Klammerausdrücke, <math>a^n b^n</math> im Vergleich zu <math>(ab)^n</math></p>

## **Unterrichtsvorhaben Q1-V**

Thema: Maschinennahe Programmierung und Grenzen der Berechenbarkeit

**Leitfragen:** *Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?*

### **Vorhabenbezogene Konkretisierung:**

Anhand einer von-Neumann-Architektur und maschinennahen Programmen wird die prinzipielle Arbeitsweise von Computern verdeutlicht **und motiviert, warum Turingmaschinen und die Registermaschine äquivalente Berechnungsmodelle sind.**

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

**Zeitbedarf:** 12 Stunden



## Sequenzierung des Unterrichtsvorhabens:

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) einige 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>d) <b>Unterprogrammtechnik</b></p> <p>e) <b>Parameterübergabe</b></p> <p>f) <b>Rekursion</b></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>• <b>kennen und verstehen die Möglichkeiten zur Modularisierung von Programmcode, Parameterübergabe und Einsatz/Funktionsweise rekursiver Strukturen in Maschinencode (I)</b></li> <li>• untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).</li> </ul>	<p><i>Beispiele:</i></p> <p>Addition von 4 zu einer eingegebenen Zahl mit einem Rechnermodell</p> <p>Schleifenstrukturen, Verzweigungen, Summe der ersten n Zahlen, Fakultät, ggT-Berechnung, Potenzfunktionen, Primzahltest... → Fachschaftspool</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 – Von-Neumann-Architektur und maschinennahe Programmierung</p> <p>(<a href="#">Download Q2-III.1</a>)</p> <p>„youtube“-Videos zur von-Neumann-Architektur / fetch-decode-execute-Zyklus</p> <p>DC – didaktischer Computer (<a href="http://www.gierhardt.de/informatik/dc/index.html">http://www.gierhardt.de/informatik/dc/index.html</a>)</p> <p>Oder <i>MikroOne</i> unter</p>

		<a href="http://mikroone.azurewebsites.net/home/">http://mikroone.azurewebsites.net/home/</a>
<p><b>2. Grenzen der Automatisierbarkeit</b></p> <ul style="list-style-type: none"> <li>a) Vorstellung des Berechenbarkeitsbegriffs, Unterscheidung von Blick auf NP-schwere aber lösbare Probleme.</li> <li>b) Vorstellung der Collatz-Funktion</li> <li>c) Vorstellung des Halteproblems</li> <li>d) Unlösbarkeit des Halteproblems</li> <li>e) Beurteilung des Einsatzes von Informationssystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</li> </ul>		<p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem</p> <p>(<a href="#">Download Q2-III.2</a>)</p> <p><i>Material aus Fachschaftspool</i></p>

## **Unterrichtsvorhaben Q2-I (im LK als Q1IIIb integriert)**

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?*

### **Vorhabenbezogene Konkretisierung:**

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt.

Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse BinaryTree (der 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 thematisiert werden - Tiefendurchlauf (*deepening depth-first search, DDFS*) (Pre-, Post- und Inorder) sowie Breitendurchlauf ( Breadth-First-Search, BFS) – und **iterativ** sowie rekursiv implementiert werden. **Bei der iterativen Implementation wird wieder zurückgegriffen auf die ADT's aus Q1 (Stack für Tiefendurchlauf, Queue für Breitendurchlauf).** Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen.

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 (der 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.

**Zeitbedarf:** 24 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Analyse von Baumstrukturen in verschiedenen Kontexten</b></p> <p>(a) Grundlegende Begriffe (Wurzel, Elternknoten, Kindknoten, Blatt, Pfad, Grad, Tiefe, Höhe, Ebene, Inhalt, Teilbaum, 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> </ul>	<p><i>Vorschlag:</i></p> <p><b>Einstieg über Spielbäume</b> als Instrument für Spielalgorithmen (Nim-Spiel, Tic-Tac-Toe) <i>siehe auch Buch: Informatik 2, Schöningh</i></p> <p>oder</p> <p>Leitprogramm der Eth Zürich <a href="http://www.educ.ethz.ch/unt/um/inf/ad/baeume2/leitprogramm.pdf">www.educ.ethz.ch/unt/um/inf/ad/baeume2/leitprogramm.pdf</a></p> <p><b>Implementation:</b></p> <p><i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum – Die Simpsons (siehe Buch Informatik 2, Schöningh)</p> <p><b>Binäre Suchbäume</b> (sortierte Speicherung von Daten) 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</p>

	<ul style="list-style-type: none"> <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 auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>• modifizieren Algorithmen und Programme (I),</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• testen Programme systematisch anhand von Beispielen (I),</li> <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> </ul>	<p>rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p><i>Beispiel:</i> Kochbuch mit alphabetisch sortierten Rezepten (Buch Informatik 2, Schöningh)</p> <p><i>Oder</i></p> <p><i>Beispiel:</i> Entscheidungsbäume</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><i>oder</i></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>Materialien:</i></p> <p>Abiturklassen, Buch Informatik 2, Schöningh, weiteres Onlinematerial</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><i>Beispiel: Informatikerbaum als binärer Baum</i></p> <p>In einem <i>binären Baum</i> 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><i>Materialien:</i></p> <p>Abiturklassen, Buch Informatik 2, Schöningh, weiteres Onlinematerial</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 Eigen-</p>		<p><i>Beispiel: Informatikerbaum als Suchbaum</i></p> <p>In einem binären <i>Suchbaum</i> 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,</p>

<p>schaften 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 Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p> <p>(e) Laufzeitbetrachtung bei binären Suchbäumen (logarithmisch), Entartung zur Liste im worst-case</p>		<p>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. Mögliche Vertiefung:</b> Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p> <p>AVL-Bäume als Beispiel für ausgeglichene Bäume und Verbesserung der Laufzeiten bei binären Suchbäumen (Vermeidung der Entartung zur Liste)</p>		<p><i>Beispiel:</i> Huffman-Codierung</p> <p><i>Materialien:</i> Erläuterungen mit Animation</p> <p><a href="http://www.iti.fh-flensburg.de/lang/algorithmen/code/huffman/huffman.htm">http://www.iti.fh-flensburg.de/lang/algorithmen/code/huffman/huffman.htm</a></p>
<p><b>5. Graphen</b></p> <p>(a) Grundlegende Begriffe (Knoten, Kanten, gerichtet, ungerichtet, Kno-</p>		<p><b>Einstieg</b> über Alltagsbeispiel Navigationssystem</p> <p>Darstellung von gerichteten Graphen , ungerichteten Gra-</p>

<p>tenmenge, Kantenmenge, Kantenzug, Kantenfolge, Weg, zyklus)</p> <p>(b) <i>Graphenprobleme</i> (z.B. Königsberger Brückenproblem (Eulerscher Zyklus), Hamiltonsche Wege und Zyklen, Traveling Salesman Problem, Knotenfärbung)</p> <p>(c) Darstellung von Graphen: Adjazenzmatrix, Adjazenzliste und ihre Modellierung</p> <p>(d) Algorithmen: Dijkstra – Kürzeste Wege</p> <p><b>Vertiefung:</b>  Weitere Algorithmen zur Abgrenzung zum Dijkstra:  - minimale Spannbäume (Kruskal, Prim)  - Bellmann-Ford-Algorithmus  - Floyd-Warshall-Algorithmus</p>		<p>phen – Übungen zu Begriffen, Definitionen</p> <p><b>Berühmte Graphenprobleme</b> theoretisch besprechen.</p> <p><i>Material: Buch „Graphen für Einsteiger – Rund um das Haus vom Nikolaus“, Manfred Nitzsche</i></p> <p>Modellierung von Graphen, Adjazenzlisten, Adjazenzmatritzen</p> <p>Implementierung von Adjazenzlisten, Adjazenzmatritzen, ebenso Graphdurchlauf iterativ, rekursiv, Dijkstra Algorithmus</p> <p><b>Material:</b></p> <p><i>Abiturklassen,  Buch Informatik 2, Schöningh  Buch Abenteuer Informatik, Gallenbacher, „Das Brückenbauproblem“</i></p> <p>Dijkstra: <a href="http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo7.php">http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo7.php</a></p> <p>Bellmann-Ford: <a href="https://www-m9.ma.tum.de/material/de/spp-bellman-ford/">https://www-m9.ma.tum.de/material/de/spp-bellman-ford/</a></p> <p>Floyd-Warshall: <a href="http://www.iti.fh-flensburg.de/lang/algorithmen/graph/warshall.htm">http://www.iti.fh-flensburg.de/lang/algorithmen/graph/warshall.htm</a></p>
---	--	--



## ***Unterrichtsvorhaben Q2-II***

Thema: Kommunikation, Sicherheit und Datenschutz in Netzstrukturen

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

Inhaltliche Schwerpunkte:

1. Einzelrechner und Rechnernetzwerke
  - (a) Schichtenmodelle
  - (b) Netzwerk-Topologien
2. Sicherheit
  - (a) digitale Signatur / Urheberrecht
3. Nutzung von Informatiksystemen, Wirkungen der Automatisierung
  - (a) Datenschutz

Inhaltsfelder: Inhaltsfeld 2 (Algorithmen), Inhaltsfeld 4 (Informatiksysteme), Inhaltsfeld 5 (Informatik, Mensch und Gesellschaft)

Zeitbedarf: 12 Stunden

Abspraken zur vorhabenbezogene Konkretisierung:

Ausgehend von einer Kommunikation zwischen zwei Kommunikationspartnern über eine einfache Leitung werden die Notwendigkeiten einer Datenübertragung erarbeitet. Die Schichten des TCP/IP-Schichtenmodells werden beispielgebunden erarbeitet (Basisbandübertragungsverfahren, Prüfverfahren, Vermittlungsschicht, Anwendungsprotokoll) und an einer Simulationssoftware getestet. Verschiedene Netzwerk-Topologien werden entwickelt und in Client-Server-Anwendungen simuliert. Im LK verbindlich, im GK wenn möglich, werden auf der Basis existenter Netzwerkklassen kleinere Client-Server-Anwendungen implementiert (Time-Client, Echo-Server, Chat, einfaches Netzwerkspiel,...)

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

Lernmittel / Materialien:

- Arbeitsblätter zur Einführung in Netzwerke
- Simulationsprogramm "Filius"
- Arbeitsblätter und Skript zu "Filius"

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Schichten des TCP/IP-Protokolls</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Notwendigkeiten einer Netzwerkkommunikation</li> <li>• Erarbeitung der Schichten des TCP/IP-Protokolls: Basisbandübertragung, Prüfverfahren, Routing/Vermittlungsschicht, Anwendungsprotokolle</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zur Einführung in Netzwerke</li> </ul>
<p>2. Simulation von Netzwerken / Netzwerk-Topologien</p> <ul style="list-style-type: none"> <li>• Erarbeitung der Topologien: Peer-to-Peer, Sterntopologie, Baumtopologie, Vermaschtes Netz</li> <li>• Simulation von Client-Server-Anwendungen</li> <li>• Simulation von Protokollen der Anwendungsschicht (POP3, SMTP, etc.)</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> <li>• <b>analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A),</b></li> </ul>	<ul style="list-style-type: none"> <li>• Simulationssoftware FILIUS</li> <li>• Arbeitsblätter und Skript zu FILIUS</li> <li>• dringend alte Abituraufgaben als Aufgabenpool nutzen, denn Protokolle für Sachanwendungen sind abiturspezifisch</li> <li>• Informatik Treff Bezirksregierung Düsseldorf Client Server Programmierung mit Sockets <a href="http://www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/clientserverjava.php">http://www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/clientserverjava.php</a></li> </ul>
<p>3. Programmierung einfacher Netzwerkanwendungen (im LK obligatorisch, im GK möglich)</p> <ul style="list-style-type: none"> <li>• Entwicklung eines Clients (Time-Client)</li> <li>• Entwicklung eines Servers (Echo-Server)</li> <li>• Kommunikation von Client und Server (Echo-Client)</li> <li>• Entwicklung eines einfachen Netzwerk-Spiels</li> </ul>	<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>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Ge-</li> </ul>	<ul style="list-style-type: none"> <li>• Arbeitsblätter zur Netzwerkprogrammierung</li> <li>• Templates und Ablaufdiagramme</li> <li>• Diagramme zum Kommunikationsfluss</li> </ul>

	<p>neralisieren (M),</p> <ul style="list-style-type: none"><li>• entwickeln und implementieren iterative Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (M/I),</li><li>• modifizieren Algorithmen und Programme (I),</li><li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li><li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li><li>• testen Programme systematisch anhand von Beispielen (I),</li><li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li></ul>	
--	--	--

## **Unterrichtsvorhaben Q2-III**

Thema: Kryptographie

Leitfragen: *Welche grundlegenden Ansätze gibt es in der Kryptographie und wie sicher sind sie? Wie schwer ist es aus welchen Gründen, mit bestimmten Verfahren verschlüsselte Nachrichten zu entschlüsseln? Welche Verfahren werden heute benutzt?*

### **Vorhabenbezogene Konkretisierung:**

Ausgehend von bekannten Verfahren (Cäsar und Vigenere) wird untersucht, wie mit diesen Verfahren verschlüsselte Nachrichten entschlüsselt werden können. Die Unterscheidung von symmetrischer und asymmetrischer Verschlüsselung wird heraus gearbeitet. Dabei werden der Diffie-Hellmann-Schlüsselaustausch und das RSA-Verfahren vorgestellt bzw. erarbeitet.

**Zeitbedarf:** 10 Stunden

### **Material**

- Kryptografieprogramm "Cryptool"
- Diverse Internet-Seiten
- Arbeitsblätter zu kryptografischen Verfahren

## Sequenzierung des Unterrichtsvorhabens

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<ol style="list-style-type: none"> <li>1. Wiederholung einfacher Verfahren (Cäsar, Vigenere, PlayFair).</li> <li>2. Entschlüsselung von mit Vigenere und Cäsar verschlüsselten Texten (Häufigkeitsanalyse, <b>Kasiski-Test</b>, <b>Friedmann-Test</b>)</li> <li>3. Symmetrische und Asymmetrische Verfahren</li> <li>4. Der Diffie-Hellman-Schlüsselaustausch und Falltür-Funktionen</li> <li>5. Das RSA-Verfahren: Vorgehen, <b>mathematischer Hintergrund</b> und Sicherheit</li> <li>6. <b>Bessere symmetrische Verfahren (DES, AES)</b></li> </ol>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern Eigenschaften, <b>Funktionsweisen</b> und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A)</li> <li>• untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</li> <li>• untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).</li> </ul>	<p><i>Cryptool</i></p> <p><i>Arbeitsblätter</i></p> <p><i>Internet-Seiten zum Kasiski-Test</i>  <a href="https://www.youtube.com/watch?v=Y6qimy9o3f4">https://www.youtube.com/watch?v=Y6qimy9o3f4</a> ,  <a href="https://www.bg.bib.de/FHDW/Intranet/gdi/pres/kasiski/kasiski.html">https://www.bg.bib.de/FHDW/Intranet/gdi/pres/kasiski/kasiski.html</a>)</p> <p><i>Internetseiten zu RSA auf</i> <a href="http://www.matheprisma.uni-wuppertal.de/Module/RSA/">http://www.matheprisma.uni-wuppertal.de/Module/RSA/</a></p>

## **Unterrichtsvorhaben Q2-IV**

Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

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

### **Vorhabenbezogene Konkretisierung:**

Ausgehend von einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Alternativ können auch Fragestellungen zum Üben vorgegeben werden. 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.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden.

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

**Zeitbedarf:** 25 Stunden

## Sequenzierung des Unterrichtsvorhabens

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: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL)</li> </ul> <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</li> <li>analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>bestimmen Primär- und Sekundärschlüssel (M),</li> <li>ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>modifizieren eine Datenbankmodellierung (M),</li> <li>modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>bestimmen Primär- und Sekundärschlüssel (M),</li> <li>überführen Datenbankschemata in vorgegebene Normalformen (M),</li> <li>verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>ermitteln Ergebnisse von Datenbankab-</li> </ul>	<p><i>Beispiel:</i> Weltdatenbank der VILN (Virtuellen Lehrerfortbildung Niedersachsen)</p> <p>Datenbank steht als download bereit und kann auf einen Datenbankserver importiert werden</p> <p><a href="http://egdservice.de/Ach/Info_Profil_9/SQL/01_Blatt_1.pdf">http://egdservice.de/Ach/Info_Profil_9/SQL/01_Blatt_1.pdf</a></p> <p><a href="http://www.vlin.de/kurs2_weiteres_2.html">http://www.vlin.de/kurs2_weiteres_2.html</a></p> <p><i>Onlineübungen:</i></p> <p><a href="http://sqlzoo.net/wiki/SQL_Tutorial">http://sqlzoo.net/wiki/SQL_Tutorial</a></p> <p><i>auch schönes Material: Buch: Klett Informatik 2 „Tabellenkalkulationssysteme und Datenbanken“ ab S. 96 Datenbankentwurf und ab S. 128 Tabellen auswerten</i></p> <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</p> <p><a href="http://dokumentation.videocenter.schule.de/old/video/index.html">http://dokumentation.videocenter.schule.de/old/video/index.html</a></p>

	<p>fragen über mehrere verknüpfte Tabellen (D),</p> <ul style="list-style-type: none"> <li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).</li> </ul>	<p>(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><i>Beispiel:</i> Schulbuchausleihe</p> <p>Unter <a href="http://www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php">www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php</a> (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p> <p><i>Material:</i> Buch Informatik 3 - Schöningh, SQL-Zoo Online, Buch Informatik 2 – gute Übersicht zu SQL-befehlen mit Erklärungen</p>
<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> <li>• Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>• Erläuterung und Modifizierung einer Datenbankmodellierung</li> </ul> <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p>		<p><i>Beispiel:</i> Fahrradverleih</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><i>Beispiel:</i> Reederei</p> <p>Die Datenverwaltung einer Reederei soll in einem</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> <li>• Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul>		<p>Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird.</p> <p><i>Beispiel: Schulverwaltung</i></p> <p>In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.</p> <p><i>Material: Buch Informatik 3 – Schöningh,</i>  <a href="http://www2.tcs.informatik.uni-muenchen.de/lehre/lehrausbildung/db_ermmodell.pdf">http://www2.tcs.informatik.uni-muen-  chen.de/lehre/lehrausbildung/db_ermmodell.pdf</a></p> <p><a href="http://projekte.gymnasium-odent-hal.de/informatik/dateien/Informatik/Jahrgangsstufe%20Q/001%20Klausuren/Themensortiert/14%20Datenbanken.pdf">http://projekte.gymnasium-  odent-  hal.de/informatik/dateien/Informatik/Jahrgangsstu-  fe%20Q/001%20Klausuren/Themensortiert/14  %20Datenbanken.pdf</a></p> <p><i>Nachschlagewerk speziell für Oberstufe Informatik:</i> <a href="http://www.flemming-">http://www.flemming-</a></p>
--	--	---

		<a href="http://universum.de/html_buch/buch.html">universum.de/html_buch/buch.html</a>
--	--	--

## **2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit**

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik 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:

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

### Fachliche Grundsätze

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

## **2.3 Grundsätze der Leistungsbewertung**

### **2.3.1 Transparenz der Leistungsbeurteilung**

Schulische Leistungsbewertung steht im Spannungsfeld pädagogischer und gesellschaftlicher Zielsetzung.

Unter pädagogischen Gesichtspunkten hat sie vornehmlich das Individuum im Blick. Hier soll sie über den Leistungszuwachs rückmelden und dadurch die Motivation für weitere Anstrengungen erhöhen. Sie ermöglicht den Schülerinnen und Schülern ihre noch vorhandenen fachlichen Defizite wie auch ihre Stärken und Fähigkeiten zu erkennen um dadurch ein realistisches Selbstbild aufzubauen. Sie ist Basis für gezielte individuelle Förderung.

Für die Erziehungsberechtigten sind Noten eine einfache und zentrale Information zum Leistungsstand ihre Kinder. Sie bieten den Anlass, über die Ursache von Defiziten und über die Beseitigung von Lernschwierigkeiten verschiedenster Art Rücksprache zu halten. Noten sind zudem Grundlage und Anlass, in den halbjährlich stattfindenden pädagogischen Konferenzen über die Schwierigkeiten und besonderen Probleme einzelner Schüler wie auch Klassen zu beraten und Maßnahmen zur Verbesserung zu beschließen.

Schulische Leistungsbewertung ist eingebettet in die durch das Schulgesetz § 48 (Grundsätze der Leistungsbewertung), APO - GOST §13 bis §17 sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe vorgegebene Grundsätze und Verfahren. Daraus erwächst für die Schulen konkret die Aufgabe, sowohl die individuellen Schwächen und Stärken der Schüler zu diagnostizieren und gegebenenfalls die Defizite durch gezielte Maßnahmen zu beseitigen sowie besondere Begabungen zu fördern.

Die gesellschaftliche Funktion von Noten zu erfüllen ist der Schule aufgegeben. Noten entscheiden mit über Schullaufbahnen, Versetzungen und Abschlüsse. Zeugnisse sind mit entscheidender Parameter bei der Zuteilung von Berufs- und Lebenschancen. Daraus erwachsen für die Beurteilenden eine besondere Verantwortung und die Pflicht einer größtmöglichen Objektivität bei der Notenfindung.

Die Fachkonferenz Informatik legt die Kriterien für die Leistungsbeurteilung fest. Die Lehrerinnen und Lehrer machen diese Kriterien den Schülerinnen und Schülern transparent.

## **2.3.2 Grundsätze der Leistungsbeurteilung**

Es gelten folgende Grundsätze der Leistungsbewertung:

- Lernerfolgsüberprüfungen sind ein kontinuierlicher Prozess. Bewertet werden alle im Zusammenhang mit dem Unterricht erbrachten Leistungen (schriftliche Arbeiten, mündliche Beiträge, praktische Leistungen).
- Leistungsbewertung bezieht sich auf die im Unterricht geförderten Kompetenzen.
- Die Lehrperson gibt den Schülerinnen und Schülern im Unterricht hinreichend Gelegenheit, die entsprechenden Anforderungen der Leistungsbewertung im Unterricht in Umfang und Anspruch kennenzulernen und sich auf sie vorzubereiten.
- Bewertet werden der Umfang, die selbstständige und richtige Anwendung der Kenntnisse, Fähigkeiten und Fertigkeiten sowie die Art der Darstellung.

## **2.3.3 Formen der Leistungsüberprüfung**

### **2.3.3.1 Kursarbeiten bzw. Klausuren**

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

Klausuren abhängig von den kontinuierlich ansteigenden Anforderungen entsprechend dem Lehrplan.

Es ist darauf zu achten, dass nicht nur die Richtigkeit der Ergebnisse und die inhaltliche Qualität, sondern auch die angemessene Form der Darstellung unabdingbare Kriterien der Bewertung der geforderten Leistung sind.

Am Couven-Gymnasium werden die Kursarbeiten in der Regel nach einem vorab festgelegten Punkteschema bewertet. Dabei unterscheiden wir zwischen der Einführungsphase (EF) und der Qualifikationsphase (Q1 und Q2). Während in der EF eine glatt ausreichende Leistung erst ab ca. 50 % gegeben wird, ansonsten äquidistante Verteilung, ist eine glatt ausreichende Leistung in der Q1 und Q2 bei 45% der Punktzahl erreicht worden. Die übrigen Notenstufen ergeben sich dann dadurch, dass für jede Notenstufe Intervalle der erreichten Punkte gebildet werden, die in der Regel gleich groß sind:

EF (wie Sek I):

ab %	Zensur
0,00%	6
25,00%	5
50,00%	4
63,00%	3
75,00%	2
88,00%	1

Q1 und Q2:

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

Die Fachkonferenz legt die Dauer der Kursarbeiten und Klausuren fest. Am Couven-Gymnasium gelten für die Sekundarstufe I und II folgende Regelungen:

Klasse	1. Klausur, 1. HJ	2. Klausur 1. HJ	1. Klausur 2. HJ	2. Klausur, 2. HJ
<b>EF</b>	---	90 min	90 min	---
<b>Q1 GK</b>	90 min	90 min	90 min	90 min
<b>Q1 LK</b>	135 min	135 min	180 min	180 min
<b>Q2 GK</b>	135 min	135 min	180 min	180 min
<b>Q2 LK</b>	225 min	225 min	255 min	255 min

In der Qualifikationsphase I kann die erste Klausur im 2. Halbjahr durch eine Facharbeit ersetzt werden.

### 2.3.3.2 Mitarbeit im Unterricht

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

Bei den mündlichen Leistungen im Unterricht sind zu bewerten:

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

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

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

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

### **2.3.3.3 Individuelle Förderung**

Die Lehrerinnen und Lehrer beobachten die individuellen Leistungen in allen Bereichen der Informatik über einen längeren Zeitraum, um auf dieser Grundlage ein Leistungsbild zu erhalten. Neben der Orientierung an den Kompetenzstandards der jeweiligen Jahrgangsstufe kann bei der Leistungsbewertung auch die jeweilige Entwicklung des Schülers bzw. der Schülerin, gemäß der zu beobachtenden Lern- und Denkfortschritte, berücksichtigt werden.

Der Informatikunterricht lebt von der verantwortungsvollen und selbständigen Arbeit der Schülerinnen und Schüler, so dass die Lehrperson die nötige Zeit hat, bei Bedarf gezielt und individuell zu fördern.

Leistungsstärkere Schülerinnen und Schüler können ihr Wissen anhand von vertiefenden Problemstellungen erweitern.

### **2.3.3.4 Bildung der Zeugnisnote**

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

## **2.4 Lehr- und Lernmittel**

Eingesetzte Lehrbücher und Arbeitsmaterialien:

- Bücher der Reihe „Informatik 2 und 3“, evtl. demnächst auch Informatik 1 (neu), vom Schöningh-Verlag
- Skripte, Arbeitsblätter, Leitprogramme, Lernvideos (z.B. joy of code)
- Umfangreiche Skripte, Leitprogramme werden digital auf Fronter bereitgestellt.
- Siehe Unterrichtsvorhaben

Eingesetzte Software (jeweils in der aktuellen Version):

- Java SDK

- BlueJ
- Greenfoot
- Java-Editor (PC) bzw. Eclipse (Mac)
- DC-Registriermaschinen-Simulationsprogramm (<http://www.oberstufeninformatik.de/dc/>)
- JFlap
- Xampp bzw. serverToGo
- Cryptool
- diverse Demonstrationsprogramme

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

Zusammenarbeit mit anderen Fächern

Fach- und aufgabenfeldbezogene sowie übergreifende Absprachen, z. B. zur Arbeitsteilung bei der Entwicklung crosscurricularer Kompetenzen (ggf. Methodentage, Projektwoche, Facharbeitsvorbereitung, Schulprofil...)

Nutzung außerschulischer Lernorte (Infosphere der RWTH, DLR-Schoollab, Rechenzentrum RWTH). In der EF wird darauf abgezielt, das Fach im Bereich des Projektmanagements mit einem Angebot der Firma Heinen Automation Monschau zu verknüpfen. Hierbei werden Lego-EV3-Roboter zu einer Fragestellung mit Java programmiert, Ergebnis und Prozess dokumentiert und vor einer Jury vorgestellt.

### **4 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 zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Jeweils vor Beginn eines neuen Schuljahres, d.h. erstmalig nach Ende der Einführungsphase im Sommer 2015 werden in einer Sitzung der Fachkonferenz für die nachfolgenden Jahrgänge zwingend erforderlich erscheinende Veränderungen diskutiert und ggf. beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird eine Arbeitsgruppe aus den zu diesem Zeitpunkt in der gymnasialen Oberstufe unterrichtenden Lehrkräften auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.