

# Informatik: Schulinterner Lehrplan für die gymnasiale Oberstufe am Anno-Gymnasium Siegburg

Das vorliegende Dokument orientiert sich an dem Beispiel für schulinterne Lehrpläne im Fach Informatik<sup>1</sup> und wurde für das Anno-Gymnasium in Siegburg ausgestaltet.

## Inhaltsverzeichnis

<b>1 Fachgruppe Informatik am Anno-Gymnasium Siegburg</b>	<b>2</b>
<b>2 Entscheidungen zum Unterricht</b>	<b>2</b>
2.1 Unterrichtsvorhaben . . . . .	3
2.1.1 Übersichtsraster Unterrichtsvorhaben . . . . .	4
2.1.2 Konkretisierte Unterrichtsvorhaben . . . . .	13
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit . . . . .	35
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung . . . . .	36
2.3.1 Beurteilungsbereich Klausuren . . . . .	36
2.3.2 Beurteilungsbereich »Sonstige Mitarbeit« . . . . .	37
<b>3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen</b>	<b>40</b>
<b>4 Qualitätssicherung und Evaluation</b>	<b>40</b>

---

<sup>1</sup>(SILP-IF-2014)

<http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/informatik/hinweise-und-beispiele/hinweise-und-beispiele.html>

# 1 Fachgruppe Informatik am Anno-Gymnasium Siegburg

Das Anno Gymnasium Siegburg ist ein voll ausgebautes vierzügiges/fünfüziges Gymnasium mit gymnasialer Oberstufe.

Bereits in der 5. und 6. Klasse wird Informatik verpflichtend gelehrt.

Weiterhin können die Schülerinnen und Schüler im Differenzierungskurs Mathematik / Informatik - kurz MI - in den Jahrgangsstufen 9 und 10 einen entsprechenden Schwerpunkt setzen.

Viele Kompetenzen werden dort bereits angebahnt, welche in der Oberstufe, gemäß des Spiralcurriculums, erweitert werden können. Es sei jedoch angemerkt, dass die Belegung des Wahlpflichtkurses MI keine Voraussetzung für die Kurswahl Informatik in der Oberstufe darstellt.

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

Der verantwortungsvolle Umgang mit Informatiksystemen durch Schülerinnen und Schüler ist als Erziehungs- und Bildungsziel im Schulprogramm ausdrücklich verankert.

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

Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor. Das Fach wird in der Sekundarstufe II im naturwissenschaftlichen Bereich angerechnet und kann sowohl mündlich als auch schriftlich gewählt werden. Seit dem Schuljahr 2020/21 bieten wir einen Leistungskurs an.

Das Anno Gymnasium besitzt drei Computerräume. Eine 1 zu 1 Relation zwischen Informatiksystemen und Schülerinnen und Schülern ist überwiegend gewährleistet.

## 2 Entscheidungen zum Unterricht

Die nachfolgend dargestellte Umsetzung der verbindlichen Kompetenzerwartungen des Kernlehrplans findet auf zwei Ebenen statt. Das **Übersichtsraster** gibt den Lehrkräften einen raschen Überblick über die laut Fachkonferenz verbindlichen Unterrichtsvorhaben pro Schuljahr. In dem Raster sind außer dem Thema des jeweiligen Vorhabens das schwerpunktmäßig damit verknüpfte Inhaltsfeld bzw. die Inhaltsfelder, inhaltliche Schwerpunkte des Vorhabens sowie Schwerpunktkompetenzbereiche ausgewiesen. Die **Konkretisierung**

der **Unterrichtsvorhaben** führt weitere Kompetenzerwartungen auf und verdeutlicht vorhabenbezogene Absprachen.

#### Hinweise zur Sequenzierung der Vorhaben

Die hier aufgeführte Sequenzierung der Unterrichtsvorhaben stellt eine Orientierung für die Lehrkraft dar. Die Reihenfolge kann ggf. angepasst werden.

#### Hinweise zur Wahl der Programmiersprache

Der Kernlehrplan schreibt keine Programmiersprache vor. Die Fachgruppe hat sich auf Java als bevorzugte Programmiersprache geeinigt. Dies scheint im Hinblick auf das Zentralabitur sowie die Objektorientierung sinnvoll.

## 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« (Abschnitt 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 sowie Schülerinnen und Schülern einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z. B. Praktika, Kursfahrten o. ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum »Übersichtsraster Unterrichtsvorhaben« zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung »konkretisierter Unterrichtsvorhaben« (Abschnitt 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von

unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Abschnitten 2.2 bis 2.3 zu entnehmen sind. Ebenso haben Schülerinnen und Schüler so die Möglichkeit, den Lehr- und Lernprozess besser zu durchdringen und selbst zu planen.

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**

Bei den Übersichten werden die Bezeichnungen für die Inhaltsfelder in der Überschrift komplett ausgewiesen, auch wenn in den Unterrichtsvorhaben nur Teilaspekte behandelt werden.

## 1) Einführungsphase

Einführungsphase	
<p><b>Unterrichtsvorhaben E-1</b></p> <p><b>Thema:</b> Informatik – Fachgebiete, typische Problemstellungen, Nutzung von Informatiksystemen und Geschichte</p> <p><b>Kompetenzbereiche (Prozesse):</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Informatiksysteme</li><li>• Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Betriebssysteme: Dateisystem</li><li>• Informatiksysteme</li><li>• Automatisierung durch Formalisierung</li><li>• Geschichte der Informatik</li><li>• Digitalisierung</li><li>• Struktureller Aufbau und Arbeitsweise singulärer Rechner am Beispiel der Von-Neumann-Architektur</li></ul> <p><b>Zeitbedarf:</b> 10 Unterrichtsstunden</p>	<p><b>Unterrichtsvorhaben E-2</b></p> <p><b>Thema:</b> Grundlagen der objektorientierten Modellierung</p> <p><b>Kompetenzbereiche (Prozesse):</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Formale Sprachen und Automaten</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und ihre Darstellung</li><li>• Syntax und Semantik einer Programmiersprache</li></ul> <p><b>Zeitbedarf:</b> 10 Unterrichtsstunden</p>

## Einführungsphase

### Unterrichtsvorhaben E-3

**Thema:** Grundlagen der Implementation

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache
- Dateisystem
- Einsatz von Informatiksystemen

**Zeitbedarf:** 20 Unterrichtsstunden

### Unterrichtsvorhaben E-4

**Thema:** Kontrollstrukturen

**Kompetenzbereiche (Prozesse):**

- Argumentieren
- Modellieren
- Implementieren

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache

**Zeitbedarf:** 15 Unterrichtsstunden

## Einführungsphase

### Unterrichtsvorhaben E-5

**Thema:** Strukturierung und Organisation von Daten, Algorithmusbegriff, Suchen und Sortieren

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

- Algorithmen
- Daten und ihre Strukturierung

**Inhaltliche Schwerpunkte:**

- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Algorithmen zum Suchen und Sortieren
- Objekte und Klassen

**Zeitbedarf:** 15 Unterrichtsstunden

### Unterrichtsvorhaben E-6

**Thema:** Klassen und ihre Implementierung

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache
- Dateisystem
- Einsatz von Informatiksystemen

**Zeitbedarf:** 20 Unterrichtsstunden

Summe Einführungsphase: 90 Unterrichtsstunden

## II) Qualifikationsphase (Q1 und Q2)

Qualifikationsphase	
<p><b>Unterrichtsvorhaben Q1-1</b></p> <p><b>Thema:</b> Verwaltung von Daten in linearen Datenstrukturen</p> <p><b>Kompetenzbereiche (Prozesse):</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</li><li>• Informatiksysteme</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Analyse, Entwurf und Implementierung von Algorithmen</li><li>• Algorithmen in ausgewählten informatischen Kontexten</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Nutzung von Informatiksystemen</li></ul> <p><b>Zeitbedarf:</b> 15 Unterrichtsstunden</p>	<p><b>Unterrichtsvorhaben Q1-2</b></p> <p><b>Thema:</b> Suchen und Sortieren auf linearen Datenstrukturen</p> <p><b>Kompetenzbereiche (Prozesse):</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</li><li>• Informatiksysteme</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Analyse, Entwurf und Implementierung von Algorithmen</li><li>• Algorithmen in ausgewählten informatischen Kontexten</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Nutzung von Informatiksystemen</li></ul> <p><b>Zeitbedarf:</b> 15 Unterrichtsstunden</p>



## Qualifikationsphase

### Unterrichtsvorhaben Q1-3

**Thema:** Bäume

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 15 Unterrichtsstunden

### Unterrichtsvorhaben Q1-4

**Thema:** Graphen

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 5 Unterrichtsstunden

## Qualifikationsphase

### Unterrichtsvorhaben Q1-5

**Thema:** Vielfalt beim Einsatz von Klassen durch Abstrakte Klassen, Polymorphie und MVC

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 15 Unterrichtsstunden

### Unterrichtsvorhaben Q1-6

**Thema:** Datenbanken

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

- Datenbanken
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen
- Sicherheit
- Wirkung der Automatisierung

**Zeitbedarf:** 15 Unterrichtsstunden

Summe Qualifikationsphase 1: 80 Unterrichtsstunden

## Qualifikationsphase

### Unterrichtsvorhaben Q2-1

**Thema:** Endliche Automaten und formale Sprachen

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

- Formale Sprachen und Automaten

**Inhaltliche Schwerpunkte:**

- Endliche Automaten *und Kellerautomaten*
- Grammatiken regulärer Sprachen *und kontextfreier Sprachen*
- *Scanner, Parser und Interpreter für eine reguläre Sprache*
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

**Zeitbedarf:** 20 Unterrichtsstunden

### Unterrichtsvorhaben Q2-2

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

**Kompetenzbereiche (Prozesse):**

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

**Inhaltsfelder:**

- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache
- Einzelrechner und Rechnernetzwerke
- Nutzung von Informatiksystemen
- Wirkung der Automatisierung

**Zeitbedarf:** 20 Unterrichtsstunden

Unterrichtsvorhaben Q2-3

**Thema:** Sicherheit in der Informatik:  
Verschlüsselung und ihre Folgen

**Kompetenzbereiche (Prozesse):**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen
- Sicherheit
- Wirkung der Automatisierung
- Grenzen der Automatisierung

**Zeitbedarf:** 20 Unterrichtsstunden

## 2.1.2 Konkretisierte Unterrichtsvorhaben

Für die Arbeit in den Informatikkursen der gymnasialen Oberstufe werden Richtlinien zur Dokumentation von Ergebnissen entwickelt. Diese werden in der eigenständigen Materialsammlung gepflegt.

In allen Konkretisierungen werden die Kompetenzen zum Bereich Kommunizieren und Kooperieren vertieft. Um die Lesbarkeit des Dokumentes zu erhöhen, werden sie bei den Unterrichtsvorhaben nicht gesondert aufgeführt.

### Einführungsphase

#### Unterrichtsvorhaben EF – 1

**Thema:** Informatik – Fachgebiete, typische Problemstellungen, Nutzung von Informatiksystemen und Geschichte

Das Fach Informatik wird in der Sekundarstufe I nur in der 5. und 6. Klasse verpflichtend unterrichtet. Die dokumentierten Vorstellungen von Informatik (nicht nur) bei Schülerinnen und Schülern sind nicht nur vielfältig, sondern erweisen sich häufig als fehlerhaft. Eine Folge sind Fehlwahlen bei Informatikstudierenden, die bis zum Studienabbruch führen können.

Daher ist eine fachliche Klärung unabdingbar. Sie betrifft zum Einen die Darstellung der sechs Fachgebiete der Informatik und zum Anderen die Arbeit an ausgewählten typischen Problemen der Informatik mit dem Ziel, die Breite der Wissenschaft Informatik für die Schülerinnen und Schülern zugänglich zu machen.

Anhand von wichtigen Persönlichkeiten wird die Geschichte der Informatik lebendig und den Schülerinnen und Schülern erfahrbar.

Viele Fachkonzepte, die in der Schulinformatik thematisiert werden, können als fundamentale Ideen der Informatik auf eine lange Geschichte in der Fachwissenschaft zurückblicken. Die geschichtliche Dimension der Entwicklung der Konzepte und Strategien, mit denen die Informatik in alle Wissens- und Lebensbereiche erfolgreich vorstoßen konnte, verdeutlichen einerseits die prinzipiellen Möglichkeiten geeigneter Abstraktionsmechanismen.

Diese Abstraktionsmechanismen wurden und werden von der Fachwissenschaft Informatik entwickelt und in Form von Informatiksystemen bereitgestellt. Nahezu alle Bereiche des menschlichen Lebens können durch geeignete Daten und Algorithmen repräsentiert werden.

## Sequenzierung des Unterrichtsvorhabens

### Kompetenzen

#### Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (IF1, M).
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (IF2, M).
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (IF4, K).
- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der Von-Neumann-Architektur (IF4, A).
- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (IF5, A).
- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (IF5, A).
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (IF5, K).
- stellen ganze Zahlen und Zeichen in Binärcodes dar (IF4, D).
- interpretieren Binärcodes als Zahlen und Zeichen (IF4, D).

#### Unterrichtssequenzen

1. Klärung des Begriffs Informatik im Hinblick auf Automatisierung, Miniaturisierung und allgegenwärtige Computer
2. Einordnung von informatischen Meilensteinen in den geschichtlichen Kontext anhand von wichtigen Persönlichkeiten
3. Erarbeitung der Fachgebiete der Informatik anhand von exemplarischen Problemen
4. Herleitung allgemeiner Dateisystemstrukturen aus gegebenen Beispielen
5. Erarbeitung und Veranschaulichung einer Von-Neumann-Architektur durch den Einsatz eines Simulators und dem Entwickeln von kurzen Programmen zur Berechnung einfachster mathematischer Probleme (Addition, Multiplikation)
6. Verwendung des Binärsystems zur Darstellung von (ganzen) Zahlen/Zeichen und Erarbeitung der schriftlichen Addition im Binärsystem

## Unterrichtsvorhaben EF – 2

**Thema:** Grundlagen der objektorientierten Modellierung

»Ausgangspunkt im Informatikunterricht ist häufig ein Problem mit lebenswelt-

lichem Bezug. Schülerinnen und Schüler erwerben und erweitern in der aktiven Auseinandersetzung mit komplexen Problemstellungen Kompetenzen, die sie zum **selbstständigen informatischen Problemlösen** befähigen. Mit der Aneignung von Strategien und Techniken zur strukturierten Zerlegung im Problemlöseprozess, zur Algorithmisierung von Abläufen sowie zur formalsprachlichen, grafischen oder symbolischen Beschreibung von Sachverhalten und Zusammenhängen erwerben die Lernenden Kompetenzen zur Bewältigung von Komplexität. Die Konstruktion eines abstrakten Modells zu einer anwendungsbezogenen Problemstellung fördert das Abstraktionsvermögen sowie kreatives und strukturelles Denken.

Die Umsetzung eines informatischen Modells in ein lauffähiges Informatiksystem hat für Schülerinnen und Schüler nicht nur einen hohen Motivationswert, sondern ermöglicht ihnen auch die eigenständige Überprüfung der Angemessenheit und Wirkung des Modells im Rückbezug auf die Problemstellung« (KMNW\_KLP\_IF\_2013).

»In der Einführungsphase werden ausgehend von einfachen Fragestellungen und unter Anleitung der Lehrperson zunächst einzelne Stufen eines Problemlösungsprozesses durchlaufen« (KMNW\_KLP\_IF\_2013).

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

#### **Grafische Beschreibungselemente:**

- Objektkarten
- Objektdiagramme
- Klassenkarte
- Klassendiagramme

Konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler werden analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Grundlegende Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektkarten, Objektdiagramme, Klassenkarten und Klassendiagramme werden eingeführt.

#### **Kompetenzen**

##### **Die Schülerinnen und Schüler**

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (IF1, M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (IF1, M),
- erstellen syntaktisch korrekte Bezeichner für Objekte, Attribute und Methoden (IF3, I),

- stellen den Zustand eines Objekts dar – Objektkarte (IF1, D),
- modellieren Objekte mit ihren Attributen, Attributwerten, Methoden und Beziehungen (IF1, M),
- stellen die Kommunikation zwischen Objekten grafisch dar (IF1, M),
- stellen die Ergebnisse der Modellierungsüberlegungen der objektorientierten Analyse grafisch dar – Objektkarten und Objektdiagramme (IF1, D),
- modellieren die Kommunikation zwischen Objekten (IF1, M),
- stellen die Ergebnisse der Modellierungsüberlegungen zum Ablauf der Kommunikation der Objekte grafisch dar – Sequenzdiagramme (IF1, D),
- setzen Sequenzdiagramme in die Punktnotation um (IF1, I),
- analysieren und erläutern eine objektorientierte Modellierung (IF1, A).

### Unterrichtsvorhaben EF – 3

**Thema:** Grundlagen der Implementation

Es erfolgt eine Einführung in die Programmierung mit der Programmiersprache Java. Ausgehend von der objektorientierten Modellierung werden erste Programme implementiert. Literale und Datentypen, Operatoren und Ausdrücke, Variablen und Ein- und Ausgaben werden eingeführt und jeweils an Beispielaufgaben praktisch genutzt.

#### Kompetenzen

##### Die Schülerinnen und Schüler

- ermitteln gemeinsame Attribute und Methoden verschiedener Objekte (IF1, M),
- klassifizieren Objekte nach Attributen und Methoden (IF1, M),
- identifizieren die Art von Attributwerten verschiedener Objekte und geben die Art fachgerecht an als Zeichenkette, Zahl, Wahrheitswert, Objekt, Sammlung von Objekten (IF1, M),
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (IF1, M),
- modellieren Klassen unter Verwendung von Vererbung (IF1, M),
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (IF1, D),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (IF1, D),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (IF1, I).



### **Thema:** Kontrollstrukturen

Um die bestehenden Modelle und Programme um vielfältige Abläufe erweitern zu können, müssen die Schülerinnen und Schüler in der Lage sein, Abläufe mit Hilfe von Kontrollstrukturen präzise zu beschreiben, zu modellieren und schließlich zu implementieren.

Bis jetzt war es immer so, dass die Programme Zeile für Zeile in der gegebenen Reihenfolge abgearbeitet wurden. Sehr häufig will man im Programm aber einzelne Anweisungen nur unter bestimmten Umständen ausführen oder je nachdem, was vorher geschah (z. B. anhängig von der Benutzereingabe) unterschiedliche Anweisungen ausführen. Hierfür nutzt man Verzweigungen. Verzweigungen gehören wie Schleifen zu den Kontrollstrukturen. Kontrollstrukturen sind ganz allgemein dafür da, den Ablauf des Programms zu steuern.

Dabei sollte nicht unterschätzt werden, wie viel Zeit Schülerinnen und Schüler benötigen, um ein Programm erfolgreich zu implementieren und zum Ablauf zu bringen. Da das Erfolgserlebnis eine hohe Motivation für die weitere Arbeit im Schulfach Informatik begünstigt, ist es notwendig, für diese Phase eine zeitliche Streckung zu ermöglichen – nur wenn alle Schülerinnen und Schüler diese Phase erfolgreich bewältigen, kann sie als abgeschlossen betrachtet werden.

Anschließend erfolgt die Einführung von Methoden unter dem Aspekt der Strukturierung von Programmen.

### **Sequenzierung des Unterrichtsvorhabens**

#### **Kompetenzen**

##### **Die Schülerinnen und Schüler**

- analysieren und erläutern einfache Algorithmen und Programme (IF2, A)
- modifizieren einfache Algorithmen und Programme (IF2, I)
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (IF2, M)
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (IF2, I)
- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (IF1, M),
- testen Programme schrittweise anhand von Beispielen (IF2, I)
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (IF3, I)
- interpretieren Fehlermeldungen und korrigieren den Quellcode (IF3, I)

## Unterrichtssequenzen

Die in der EF erworbenen Kenntnisse über Ablauf- und Kontrollstrukturen bilden die Grundlage für einige der Unterrichtsvorhaben in der Qualifikationsphase. Unabhängig von den gewählten Unterrichtsbeispielen erarbeiten die Schülerinnen und Schüler die nachfolgend aufgeführten Ablauf- und Kontrollstrukturen:

- IF-Anweisungen (auch verschachtelte)
- WHILE-Schleifen und/oder FOR-Schleifen

Die Schülerinnen und Schüler sollen die aufgeführten Ablauf- und Kontrollstrukturen auf verschiedenen Ebenen erfassen und beschreiben können:

1. Verbale (umgangssprachliche) Formulierung des Ablaufs
2. Struktogrammdarstellung
3. Programmquelltext in der gewählten Programmiersprache

Da die Schülerinnen und Schüler Vorwissen zu Kontrollstrukturen mitbringen und zumindest ohne Fachsprache Abläufe umgangssprachlich verbal beschreiben können, bietet es sich an, die obige Bearbeitungsreihenfolge einzuhalten, also von der Umgangssprache zu Struktogrammen und Programmquelltexten.

## Unterrichtsvorhaben EF – 5

**Thema:** Strukturierung und Organisation von Daten, Algorithmusbegriff, Suchen und Sortieren

Die Datenstruktur Array (Feld) wird eingeführt und zur Speicherung von gleichartig strukturierten Daten genutzt.

Die Suchverfahren Lineare Suche und die Binäre Suche werden an einem Beispielkontext erarbeitet. Sortierverfahren werden besprochen und im Hinblick auf das Laufzeitverhalten verglichen.

### Sequenzierung des Unterrichtsvorhabens

#### Kompetenzen

#### Die Schülerinnen und Schüler

- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (IF2, D)
- entwerfen einen weiteren Algorithmus zum Sortieren (IF2, M)

- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (IF2, A)
- testen Programme schrittweise anhand von Beispielen (IF2, I)
- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (IF1, M)

## Unterrichtsvorhaben EF – 6

### **Thema:** Klassen und ihre Implementierung

Bei der objektorientierten Programmierung wird das Programm in voneinander unabhängige Teile gekapselt (Objekte), die über im Vorfeld festgelegte Schnittstellen miteinander kommunizieren. Es wird von Objekten und ihren Beziehungen abstrahiert und zu Klassen generalisiert.

### **Kompetenzen**

#### **Die Schülerinnen und Schüler**

- ermitteln gemeinsame Attribute und Methoden verschiedener Objekte (IF1, M),
- klassifizieren Objekte nach Attributen und Methoden (IF1, M),
- identifizieren die Art von Attributwerten verschiedener Objekte und geben die Art fachgerecht an als Zeichenkette, Zahl, Wahrheitswert, Objekt, Sammlung von Objekten (IF1, M),
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (IF1, M),
- modellieren Klassen unter Verwendung von Vererbung (IF1, M),
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (IF1, D),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (IF1, D),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (IF1, I).

### **Unterrichtssequenzen**

#### **Klasse als Abstraktion**

1. Die Objekte eines Szenarios werden gruppiert und ihre Methoden und Attribute verglichen.
2. Für die Klassen werden Klassenkarten erstellt, die die Methoden und Attribute aus der Vereinigung der gruppierten Objekte erhalten.
3. In einem Klassendiagramm werden die Klassenkarten eines Szenarios zusammengeführt und durch Beziehungen miteinander verknüpft.

### **Überführung von Klassenkarten in eine Programmiersprache**

1. Aus der Darstellungsform Klassenkarte wird der Quellcode in einer Programmiersprache erstellt.
2. Die Klassendeklaration in einer Programmiersprache wird in die Darstellungsform Klassenkarte überführt.
3. Formale Korrektheit des Quellcodes zur Klassendeklaration in einer Programmiersprache wird mittels der zugrundeliegenden Syntax geprüft/entwickelt/diskutiert.

### **Weitere Inhalte**

1. Abstrakte Klassen
2. Vererbung
3. Zusammengesetzte Datentypen
4. Kapselung
5. Assoziation

## Qualifikationsphase 1

### Unterrichtsvorhaben Q1 – 1

**Thema:** Verwaltung von Daten in linearen Datenstrukturen

Listen kommen in Realweltsituationen in allgemeiner und in spezieller Form vor. Sei es eine Warteschlange an einer Kasse, eine Polizeikontrolle oder das Stapeln von Getränkeboxen. In allgemeiner Form bei der Zusammenstellung von Zügen oder bei der Aufstellung einer Mannschaft beim Staffellauf. Eine Modellierung derartiger Abläufe stellt eine geeignete Möglichkeit dar, Listenstrukturen zu thematisieren.

#### Kompetenzen

##### Die Schülerinnen und Schüler

- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (IF1, D)
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, A),
- implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2 LK, I),
- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),
- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),
- stellen die Kommunikation zwischen Objekten grafisch dar (D),
- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D).

#### Unterrichtssequenzen

Es gibt zwei Möglichkeiten, die Listen zu thematisieren: entweder man beginnt mit der allgemeinen Form (Einfügen und Entnehmen an einer beliebigen Stelle) oder man beginnt mit einer der speziellen Formen der Liste: Schlange (Queue) oder Stapel (Stack). Bei den letztgenannten ist das Einfügen und Entnehmen nicht an einer beliebigen Stelle möglich, so dass hier von speziellen Formen der Liste gesprochen werden kann, die dem FIFO- bzw. LIFO-Prinzip unterliegen.

### Unterrichtsvorhaben Q1 – 2

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

Die Schülerinnen und Schüler haben bereits im Unterrichtsvorhaben Q1-1 die

Modellierung und Implementierung linearer Datenstrukturen kennengelernt und angewendet. Sie kennen daher den strukturellen Aufbau verketteter Listen und ihre Standardoperationen und haben sie in einfachen Beispielen verwendet. Außerdem haben die Schülerinnen und Schüler in der EF einfache Such- und Sortieralgorithmen informell (also ohne Implementierung) an realweltlichen Objekten kennengelernt. Darauf baut dieses Unterrichtsvorhaben auf.

## **Kompetenzen**

### **Die Schülerinnen und Schüler**

- analysieren und erläutern Algorithmen und Programme (IF2, A)
- modifizieren Algorithmen und Programme (IF2, I)
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (IF2, D)
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung« und »Teilen und Herrschen« (IF2, M)
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I)
- testen Programme systematisch anhand von Beispielen (IF2, I)
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, A)
- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (IF2, I)
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (IF2, A)
- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (IF3, I)
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (IF3, A)
- interpretieren Fehlermeldungen und korrigieren den Quellcode (IF3, I)
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (IF4, I)

### **Unterrichtssequenzen**

Zunächst werden Suchalgorithmen erarbeitet. Es bietet sich an, auf im vorangegangenen Unterrichtsvorhaben modellierte und implementierte Beispiele für lineare Datenstrukturen zurückzugreifen und diese ggf. noch mit unsortierten Daten zu füllen. Die Schülerinnen und Schüler entwickeln eigene Algorithmen zum Suchen von Inhalten in linearen Datenstrukturen und vergleichen diese mit Standard-Suchalgorithmen, wie die lineare Suche und die binäre Suche (hierfür ist die direkte

Adressierung praktisch Voraussetzung, daher Formalisierung und Implementierung nur im Array).

Die folgende Übersicht stellt eine mögliche Strukturierung für die Erarbeitung von Suchalgorithmen dar:

1. Entwicklung und Durchführung von Suchalgorithmen anhand realweltlicher Objekte (z. B. Doppelkopfkarten, Supertrumpf-Karten, Münzen etc.)
2. strukturierte Darstellung der Algorithmen
3. Implementierung der Algorithmen (am besten in bestehende Projekte)
4. Testen der Implementationen

Eine einfache Aufwandsbetrachtung der Sortieralgorithmen führt schnell zu der Erkenntnis, dass der Suchaufwand je nach gewähltem Algorithmus bei vorsortierten Daten geringer sein kann. Dies führt zur Notwendigkeit von Sortieralgorithmen. Auch hier entwickeln die Schülerinnen und Schüler eigene Algorithmen und vergleichen diese mit Standard-Suchalgorithmen, wie Bubble-Sort, Selection-Sort, Insertion-Sort, Quick-Sort. Die Erarbeitung der Sortieralgorithmen kann analog zur Erarbeitung der Suchalgorithmen erfolgen (siehe obige Übersicht).

## Unterrichtsvorhaben Q1 – 3

### **Thema:** Bäume

Man sagt oft, dass nichtlineares Denken bessere Leistungen hervorbringen kann. Bäume bilden eine hierarchische Struktur ab, die man im realen Leben z. B. aus Stammbäumen kennt. Strukturen von Unternehmen sind häufig hierarchisch organisiert (Aufbauorganisation). Weitere Beispiele sind das Inhaltsverzeichnis eines Buches und – aus technischer Sicht – die Struktur des Dateisystems eines Datenträgers. Hierbei handelt es sich um allgemeine Bäume. Ein Sonderfall des allgemeinen Baumes ist der Binärbaum, bei dem jeder Elternknoten höchstens zwei Kindknoten haben kann. Diese Bäume tauchen im realen Leben z. B. dort auf, wo Entscheidungen getroffen werden, die exklusiv mit »ja« oder »nein« beantwortet werden (Entscheidungsbäume). Effizientes Suchen erfolgt im allgemeinen Fall auf Grundlage einer Baumstruktur, so dass in jedem Schritt der Suche die Hälfte der Elemente ausgeschlossen werden kann. Somit ist die Anzahl der Schritte beschränkt mit:  $\mathcal{O}(\log n)$ . Dies trifft auch auf effiziente Sortierverfahren zu, die sich explizit oder implizit den Baum zu nutze machen und in  $\mathcal{O}(n \log n)$  arbeiten.

### **Kompetenzen**

#### **Die Schülerinnen und Schüler**

- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren

Aufbau (IF1, D),

- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (IF1, M),
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung« und »Teile und Herrsche« (IF2, M),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I),
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (IF2, A),
- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (IF2, I),
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (IF2, A),

### Unterrichtssequenzen

Beginnend bei einem Stammbaum, einer Unternehmensstruktur oder des Inhaltsverzeichnis eines Buches wird die Struktur eines allgemeinen Baumes hergeleitet. Nach einer allgemeinen Definition kann anhand eines Inhaltsverzeichnis herausgearbeitet werden, dass ein allgemeiner Baum auch eine Ordnungsrelation aufweisen kann. Bei einem allgemeinen Baum wird in Folge über Traversierungen gesprochen. Beginnend mit der Tiefe und Höhe des Baumes. Anhand der Fragestellung, wie alle Knoten eines Baumes »besucht« werden können, wird die Preorder-Traversierung erarbeitet. Hierbei wird deutlich, dass ein rekursives Vorgehen notwendig ist. Ausgehend von der Fragestellung, wie sich die Speicherplatzbelegung eines Datenträgers ermitteln lässt, kann die Postorder Traversierung eingeführt werden.

Binärbäume haben in der Informatik einen besonderen Stellenwert. Die Schüler kennen aus dem Mathematikunterricht bereits Termbäume. Ausgehend von einem vollständig geklammerten Ausdruck soll ein Termbaum modelliert werden. In diesem Zusammenhang wird der Baum durch Anwendung einer binären Postordermethode ausgewertet und mittels der neu einzuführenden Inorder-Traversierung wieder als der Term ausgegeben, der als Eingabe für den Baum verwendet wurde.

Wird der Binärbaum nun um eine Ordnungsrelation ergänzt, kommt man zur Anwendung eines binären Suchbaumes. Es soll schnell ein bestimmtes Objekt gefunden werden. Steht man vor einem sortierten Bücherregal so wird man ein bestimmtes Buch jedoch nicht linear suchen, sondern intuitiv in der Mitte beginnen und bei jeder Entscheidung im Laufe der Suche die Hälfte der Bücher ausschließen. Dieses Verfahren kann nun auch auf den modellierten Baum übertragen werden. Es sollte jedoch berücksichtigt werden, dass die Effizienz eines binären Suchbaums davon abhängt, wie dieser aufgebaut worden ist. Es sollte verdeutlicht werden, dass dieser zu einer linearen Liste degradieren kann, wenn die Objekte geordnet in den Baum eingefügt werden.



## Unterrichtsvorhaben Q1 – 4

**Thema:** Graphen

Dieses Vorhaben ist für den Leistungskurs verpflichtend, es wird jedoch empfohlen, die Algorithmen von Dijkstra und Kruskal im Grundkurs ohne Implementationen zu thematisieren.

Graphen sind Bestandteil vieler Anwendungen aus der realen Welt. Zu aller erst denkt man an Navigationssysteme, bei denen das Wegenetz mittels Graphen in Informatiksystemen abgebildet wird. Aber auch Beziehungsgeflechte zwischen Personen und Optimierungsprobleme in der Betriebswirtschaftslehre lassen sich mit Graphen abbilden.

### Kompetenzen

#### Die Schülerinnen und Schüler

- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (IF1, D),
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (IF2, D),
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien »Modularisierung«, »Teile und Herrsche« (IF2, M),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (IF2, I),
- erläutern Operationen dynamischer (linearer und nicht-linearer) Datenstrukturen (IF2, A),
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (IF2, A).

### Unterrichtssequenzen

Anhand eines Graphen für ein Verkehrsnetz werden die Begriffe Knoten, Kanten, gerichteter Graph, ungerichteter Graph, Eingangsgrad eines Knotens und Ausgangsgrad eines Knotens herausgearbeitet.

Anhand von Flugrouten kann das Thema kürzeste Wege besprochen werden und der Algorithmus von Dijkstra erarbeitet werden.

## Unterrichtsvorhaben Q1 – 5

**Thema:** Vielfalt beim Einsatz von Klassen durch Abstrakte Klassen, Polymorphie und MVC

Mit dem Prinzip der Modularisierung respektive Trennung von Teilbereichen haben sich die Schülerinnen und Schülern im Unterrichtsvorhaben über die linearen Datenstrukturen beschäftigt. Dort wird die Struktur vom Inhalt getrennt. Auf einem höheren Niveau werden nun komplette Klassen Kategorien zugeordnet. Die Einordnung in Klassen, die die Welt modellieren, Klassen, die für die Interaktion mit dem Benutzer zuständig sind und Klassen, die das Bindeglied darstellen und die Programmlogik übernehmen, stellt eine solche konzeptionelle Modularisierung dar.

Das Unterrichtsvorhaben kann als eigenständiges Vorhaben durchgeführt oder aber auch in anderen Vorhaben integriert werden. Im Grundkurs kann das MVC-Prinzip im Anschluss an das Unterrichtsvorhaben zu linearen Datenstrukturen durchgeführt werden, um so zu zeigen, dass Programme so entwickelt werden sollten, dass sie erweiterbar bzw. leicht veränderbar sind. Generell bietet sich hierbei an, einen Ansatz zu wählen, bei dem verschiedene Benutzerschnittstellen von den Schülerinnen und Schülern entwickelt werden. Dies können verschiedene grafische Benutzeroberflächen oder aber auch eine Konsole sein, über die der Benutzer interagiert. So wird klar, dass die Programmlogik gleich bleibt, die Ausgabe aber verschieden – z. B. je nach System – gestaltet werden kann.

In dieses Vorhaben müssen auch abstrakte Klassen und Interfaces behandelt werden, die z. B. bei der Bereitstellung von Schnittstellen genutzt werden können. Außerdem sollte die Polymorphie, also z. B. gleichnamige Methode, die je nach übergebenen Datentypen unterschiedlich reagieren, genutzt werden.

## **Kompetenzen**

### **Die Schülerinnen und Schüler**

- analysieren und erläutern objektorientierte Modellierungen (IF1, A),
- modellieren abstrakte und nicht abstrakte Klassen unter der Verwendung von Vererbung durch Spezialisierung und Generalisierung (IF1, M),
- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (IF1, M),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (IF1, I),
- analysieren und erläutern Algorithmen und Programme (IF2, A),
- modifizieren Algorithmen und Programme (IF2, I),
- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),
- stellen die Kommunikation zwischen Objekten grafisch dar (D),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- dokumentieren Klassen (D).

## Unterrichtssequenzen

Dass es für unterschiedliche Situationen sinnvoll ist, verschiedene Arten von Ausgaben (View) anzeigen zu können, ist den Schülerinnen und Schülern aus der Nutzung von Informatiksystemen bekannt. Bei den im Zusammenhang mit den bisher erstellten Programmen wurde das Hauptaugenmerk auf die fachliche Modellierung (Model) gelegt. Der interaktiven Steuerung (Control) wurde wenig Aufmerksamkeit geschenkt. Daher bietet sich ein Einstieg über die Analyse bereits erstellter Programme an, die nicht konsequent nach dem MVC-Prinzip erstellt wurden. Eine leitende Fragestellung ist »Wie kann eine anders gestaltete Oberfläche für dieses Programm realisiert werden?«. Als Beispiel für eine alternative Ausgabe kann die Konsole verwendet werden. Hierbei wird die Notwendigkeit der Einteilung der Klassen in Gruppen ersichtlich, da die Controller-Klasse die Ausgabe organisiert und die Model-Klassen sowohl bei einem Konsolen-Programm, als auch bei einem Programm mit GUI gleich bleiben. Der umgekehrte Weg ist natürlich auch möglich. Ein Programm, das auf einer Konsolenausgabe basiert, so zu modifizieren, dass ohne großen Aufwand eine GUI hinzugefügt wird. In beiden Fällen sollten mit Hilfe von abstrakten Klassen und Interfacen entsprechende Schnittstellen definiert werden.

Vor allem methodisch bieten sich hier viele Möglichkeiten. Bei einer konsequenten Entwicklung gemäß des MVC-Prinzips kann in den Gruppen je nach Leistungsniveau arbeitsteilig gearbeitet werden. Die Erstellung der Model-Klassen dürfte hierbei eher einfach sein. Bei obigem Ansatz, bei dem ein fertiges Programm modifiziert wird, können verschiedene Oberflächen parallel entwickelt werden. Die Oberflächen sollten dann einfach austauschbar sein. So lernen die Schülerinnen und Schüler, wie wichtig es ist, im Vorfeld Schnittstellen klar zu definieren.

## Unterrichtsvorhaben Q1 – 6

### **Thema:** Datenbanken

Datenbanken sind Grundlage vieler Informatiksysteme, mit denen Schülerinnen und Schüler alltäglich in Kontakt kommen. Meist merken sie nichts davon. Soziale Netzwerke, Fahrplanauskunftssysteme, Online-Shops, die Verarbeitung der Patientendaten beim Arzt usw. basieren auf Datenbanksystemen. Aus diesem Grund erscheint es sinnvoll, anhand eines Realweltproblems in die Thematik der Datenbanken einzusteigen und dieses System z. B. einen Online-Shop oder eine Supermarktkasse zu analysieren, um so den Aufbau eines Datenbanksystems kennen zu lernen. In Folge wird dieses Datenbanksystem modelliert und implementiert.

### **Kompetenzen**

#### **Die Schülerinnen und Schüler**

- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige

- Attribute, Relationen und Kardinalitäten (IF1, M),
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (IF1, D),
  - modifizieren eine Datenbankmodellierung (IF1, M),
  - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (IF1, M),
  - bestimmen Primär- und Sekundär- und Fremdschlüssel (IF1, M),
  - analysieren und erläutern eine Datenbankmodellierung (IF1, A),
  - stellen grafisch den Ablauf einer Anfrage an ein Datenbanksystem dar (Client-Server-Modell) (IF4, M),
  - erläutern die Eigenschaften normalisierter Datenbankschemata (IF1, A),
  - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (IF1, D),
  - überführen Datenbankschemata in die erste bis dritte Normalform (IF1, M),
  - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),
  - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),
  - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),
  - erläutern Eigenschaften, Funktionsweisen und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A).

### **Unterrichtssequenzen**

Gehen wir davon aus, dass die Schülerinnen und Schüler bereits objektorientiert modellieren können, kann eine unterrichtliche Sequenzierung umgesetzt werden, bei der das objektorientierte Modell über das MVC-Konzept zur ER-Modellierung verwendet werden.

Folgender Ablauf erweist sich für den Datenbankentwurf als sinnvoll:

- Datenanalyse für ein bestehendes Realweltproblem.
- Erstellung eines ER-Modells.
- Überführung des ER-Modells in ein relationales Modell.
- Einführung der Normalformen und Überprüfung des relationalen Modells auf die ersten drei Normalformen.

Bei der Datenanalyse für ein bestehendes Realweltproblem, wie etwa ein Online-Shop oder eine Kasse im Supermarkt sollen Vorteile von Datenbanksystemen gegenüber dateibasierter Datenspeicherung von den Schülerinnen und Schülern geäußert werden. Auch sollte die Client-Server-Struktur während des Anfrageprozesses erläutert werden.

Während der Modellierungsphase erlernen die Schülerinnen und Schüler das Anwenden der Fachbegriffe wie etwa Entität, Entitätstyp, Attribute, Beziehung, Kardinalität, Primärschlüssel, Sekundärschlüssel, Fremdschlüssel.

Bei der Überführung des ER-Modells in ein relationales Modell argumentieren die Schülerinnen und Schüler, welche beteiligten Attribute bei der Beziehungsrelation Primärschlüssel sein können und formulieren für die Beziehungstypen jeweils Abbildungsregeln. Sinnvoll ist das exemplarische Einfügen von Beispieldatensätzen in eine Relationentabelle und ebenso die schematische Darstellung einer Relation. Durch ein vorgegebenes Beispiel können Schwächen (Datenredundanz, Dateninkonsistenz) in einer Datenbankmodellierung den Schülerinnen und Schülern verdeutlicht werden. Durch die Analyse der Modellierungsschwächen können grundsätzliche Regeln erarbeitet werden, die später bei den drei Normalformen eine wesentliche Rolle spielen:

- Attribute können nicht sinnvoll weiter unterteilt werden (Vorname und Name sind z. B. bereits unterteilt)
- Attribute sind nicht nur von Teilen des Primärschlüssels (funktional) abhängig.
- Ein Attribut ist nicht von einem anderen Nichtprimärschlüssel-Attribut funktional abhängig.

Neben der Konstruktion der Datenbank durch die wissensbasierte Modellierung ist es wichtig, Anfragen an die Datenbank stellen zu können. Bei relationalen Datenbanken wird nach den Vorgaben für das Zentralabitur in NRW SQL für Anfragen verwendet. Dabei sind folgende Aspekte zu berücksichtigen:

- Datendefinition (Datenbank und Tabellen Anlegen sowie Einfügen der Relationen)
- Datenmanipulation (Insert, Update, Delete)
- Anfragen an die Wissensbasis
  - Abfragen auf einer Tabelle
  - Abfragen über mehrere Tabellen
  - Unterabfragen/Verschachtelungen, Sortieren, Gruppieren, Aggregationsfunktionen
  - Relationalalgebraische Betrachtung der durchgeführten Abfragen

### Unterrichtsvorhaben Q2 – 1

**Thema:** Endliche Automaten und formale Sprachen

Endlichen Automaten und formalen Sprachen gehört zum Fachgebiet Theoretische Informatik. Sie bildet einen Grundbaustein für die Berechenbarkeits- und Komplexitätstheorie. Dabei wird untersucht, welche Probleme mit Automaten gelöst werden können.

Die Transduktoren, als Teilgruppe der endlichen Automaten, haben einen Bezug zur Lebenswelt der Schülerinnen und Schülern, da mit ihnen zum Beispiel Ticket- oder Getränkeautomaten modelliert werden können. Eine besondere Rolle kommt den Akzeptoren zu. Die von ihnen akzeptierten Wörter bilden eine reguläre Sprache, die sich über Grammatiken oder reguläre Ausdrücke darstellen lässt. Anwendung finden diese Bereiche unter anderem beim Parsern, die zum Untersuchen von Texten eingesetzt werden.

#### Kompetenzen

##### Die Schülerinnen und Schüler

- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (IF3, A),
- ermitteln die Sprache, die ein endlicher Automat akzeptiert (IF3, D),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (IF3, M),
- stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (IF3, D),
- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (IF3, M),
- analysieren und erläutern Grammatiken regulärer Sprachen (IF3, A),
- modifizieren Grammatiken regulärer Sprachen (IF3, M),
- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (IF3, A),
- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (IF3, M),
- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (IF3, M),
- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (IF3, D),
- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (IF3, A).

## Unterrichtssequenzen

Der erste Teil des Unterrichtsvorhaben ist die Behandlung der endlichen Automaten. Dabei werden zuerst die Transduktoren behandelt um an die Erfahrungswelt der Schülerinnen und Schüler anzuknüpfen. Bei den anschließenden Akzeptoren sollten nicht nur die Deterministischen Automaten (DEA) behandelt werden, sondern auch die nichtdeterministischen (NEA) und ihre Umwandlungen. Eine Vielzahl von Problemen lässt sich einfacher mit einem NEA von Hand modellieren, der anschließend in einen DEA umgewandelt werden kann.

Auf den Automaten aufbauend werden Grammatiken und die Bedingungen für reguläre Grammatiken behandelt, so dass auf dieser Grundlage der Begriff der Sprache ausgearbeitet werden kann. Darauf folgen die reguläre Ausdrücke, dessen Einsatz zum Beispiel im Bereich der Suche vertieft wird.

## Unterrichtsvorhaben Q2 – 2

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

Vernetzte Informatiksysteme bestimmen die heutige IT-Infrastruktur und damit haben sie – denkt man zum Beispiel an die zahlreichen Internetanwendungen – einen nicht unerheblichen Einfluss auf unsere Lebenswelt. Zunächst sollten netzwerktechnische Grundlagen betrachtet werden. Dies gilt sowohl für den Grundkurs als auch für den Leistungskurs. Dabei erscheint es sinnvoll, folgende Aspekte zu betrachten:

- Definition eines Netzwerkes
- Netzwerktypen nach Ausdehnung (LAN, MAN, WAN, GAN)
- Peer-to-Peer vs. Client-Server-Netzwerk
- OSI-Schichtenmodell und die Funktion der Schichten
- Frames
- Netzwerktopologien (Ring-, Stern-, Bus-, Baum- und Maschen-Netz)
- Arten der Verkabelung
- Nachrichtentechnische Grundlagen (Digitalisierung, Modulationsverfahren, Übertragungsmedien)
- Übertragungsprotokolle (CSMA/CD und Token Passing)
- Netzwerkhardware

In Folge wird am Beispiel des TCP/IP-Protokolls der Protokollbegriff vertiefend betrachtet. Dies kann sinnvoller Weise in Form von Schülerreferaten erfolgen. Dabei erscheinen die folgenden Themen sinnvoll:

- Wesen eines Protokolls und Low-Level-Protokolle (MAC, LLC, SAP und SNAP)
- IP, ICMP, ARP, RARP, Routing-Protokolle und TCP

- Adressierung und Subnetze IPv4
- Dynamic Host Configuration Protokoll DHCP
- Routing Grundlagen/Routing Information Protokoll (RIP)/OSPF (Open Shortest Path First)
- Namensauflösung Host-Datei/Domain Name System (DNS)
- File Transfer Protokoll (FTP)
- Hypertext Transfer Protokoll (HTTP)
- E-Mail (SMTP/POP3/IMPAP4)
- Weiterentwicklung IPv6

Nach der eher theoretischen Betrachtung können bei ausreichend verfügbarer Zeit einige Netzwerkanwendungen unter Verwendung der Landesklassen implementiert. Dabei kann beispielhaft die folgende Reihenfolge von Anwendungen umgesetzt werden:

- Portscanner
- Echo-Client/Echo-Server
- Chat-Client/Chat-Server

Es erscheint in diesem Zusammenhang sinnvoll, den Echo-Client und Echo-Server gemeinsam mit den Schülern zu entwickeln und die Schüler dann in Folge die Erweiterung des Echo-Client und Echo-Server zum Chat-Client bzw. Chat-Server durchführen zu lassen.

Es erscheint als Sinnvoll zum Abschluss die gesellschaftlichen Auswirkungen des Einsatzes von Netzwerken zu betrachten. Hier sollten Beispiele aus der Lebenswelt der Schüler (z. B. Soziale Netzwerke) als auch Auswirkungen auf die Arbeitswelt thematisiert werden.

## **Kompetenzen**

### **Die Schülerinnen und Schüler**

- beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (IF4, A).
- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (IF5, A),
- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (IF5, A)
- analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (IF2 LK, A),



- analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (IF4 LK, A),
- entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (IF4, M).

### **Unterrichtssequenzen**

Bezugnehmend auf die oben ausgewiesenen Bereiche sollten grundlegende Definitionen zu Beginn gegeben werden. Danach sollte eine grundsätzliche Einführung von Referenzmodellen erfolgen. Hierbei ist nach Möglichkeit auf die zwei gebräuchlichsten (TCP/IP, OSI) zu verweisen. Anhand von Analogien aus der Alltagswelt (Telefonat zweier Regierungschefs inklusive Dolmetscher) kann man die verschiedenen Schichten im OSI-Modell verdeutlichen.

Danach ist die exemplarische Bearbeitung von mindestens zwei Protokollen sinnvoll, bevor die Schülerinnen und Schüler ihr eigenes Protokoll gemäß Vorgaben (vgl. Zentrale Klausuren Informatik ab 2008, NRW) entwickeln sollen. Ggf. könnten weitere Protokolle in Referate ausgelagert werden.

Ein Großteil der oben aufgeführten Themen (Routing, Topologien, DNS) lässt sich zeitbedingt nicht im Detail besprechen und sollte als Referate vergeben werden. Auch im Grundkurs sollte die Implementation von mindestens einem Beispiel der Client-Server-Kommunikation (z. B. ECHO-Client) erfolgen.

## **Unterrichtsvorhaben Q2 – 3**

**Thema:** Sicherheit in der Informatik: Verschlüsselung und ihre Folgen

Im Zeitalter vernetzter Informationssysteme ist der sichere Austausch von Daten über ein Netzwerk ein wichtiger Aspekt. Man denke an Homebankinganwendungen oder das Versenden von E-Mails die persönliche Daten beinhalten und für Dritte nicht lesbar sein sollen.

Bei der Betrachtung von Verschlüsselungsverfahren sollen symmetrische und asymmetrische, mono- und polyalphabetische Verschlüsselungsverfahren thematisiert werden. In diesem Zusammenhang scheint eine genetische Vorgehensweise sinnvoll, auch wenn diese historischen Verfahren heute aus Sicherheitsgründen in der praktischen Anwendung nicht mehr zum Einsatz kommen.

Daran schließen sich die Grundlagen der modernen Verschlüsselung, wie das RSA-Verfahren, an. Praktischen Nutzen können die Schüler durch die Verwendung von PGP/GPG bekommen, was zum verschlüsseln von Daten und besonders E-Mails eingesetzt wird. Hieran lassen sich auch verschiedene Möglichkeiten des Schlüsselaustausch exemplarisch verdeutlichen.

## **Kompetenzen**

### **Die Schülerinnen und Schüler**

- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (IF4, A).

### **Unterrichtssequenzen**

Bei der chronologischen Reihenfolge könnte ein möglicher Ablauf folgende Verfahren umfassen:

- Caesar-Chiffre
- Vigenere-Verfahren
- RSA-Verschlüsselung
- mit PGP/GPG verschlüsselte Daten

Das Caesar- und das Vigenere-Verfahren sollte auf jeden Fall implementiert werden. Zusätzlich können auch noch weitere einfache Verfahren, auch solcher für eine reine Codierung, im Vorfeld an Lernstationen erfahren werden.

Die Implementation von RSA sollte erfolgen, wenn die mathematischen und implementationstechnischen Fähigkeiten der Schüler des Grundkurses dies hergeben. Ergänzend wird die Betrachtung weiterer kryptologischer Verfahren und grundlegender Themen sinnvoll, z. B.:

- Codierungen (ASCII, Freimaurer-Alphabet)
- Hash-Verfahren (SHA1, MD5)
- Steganographie
- Schlüsselaustausch (Diffie-Hellman)

Daran schließen die Einwegverfahren wie Hashcodes MD5 und SHA1 an. Sie sollten betrachtet werden, da diese häufig bei Datenbanksystemen zum Einsatz kommen. Hier kann durch Recherche einiges über die Sicherheit der Verfahren herausgefunden werden.

Der Schlüsselaustausch stellt ein wesentliches Element zur Anwendung der Verfahren in der Praxis dar, so dass zumindest eine anschauliche Betrachtung (vgl. CS unplugged) im Unterricht erfolgen sollte.

## 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Anno-Gymnasiums fachmethodische und fachdidaktische Grundsätze beschlossen. Die Grundsätze 1 bis 13 beziehen sich auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 14 bis 20 sind fachspezifisch angelegt.

### Überfachliche Grundsätze:

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

## **Fachliche Grundsätze:**

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

## **2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung**

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

### **2.3.1 Beurteilungsbereich Klausuren**

Bei der Formulierung von Aufgaben werden die Operatoren, die auch für die Abiturprüfungen im Fach Informatik gelten, eingeführt und erklärt. In den Aufgabenstellung für die Klausuren ist auf sie zurückzugreifen.

## **Instrumente**

In den verschiedenen Jahrgangsstufen werden folgende Klausuren geschrieben.

**Einführungsphase:** eine Klausur pro Halbjahr über zwei Unterrichtsstunden

Grundkurs:

**Qualifikationsphase 1:** zwei Klausuren pro Halbjahr über zwei Unterrichtsstunden

**Qualifikationsphase 2.1:** zwei Klausuren über drei Unterrichtsstunden

**Qualifikationsphase 2.2:** Vorabiklausur über drei Unterrichtsstunden und Abiturarbeit

Leistungskurs:

**Qualifikationsphase 1:** zwei Klausuren pro Halbjahr über drei Unterrichtsstunden

**Qualifikationsphase 2.1:** zwei Klausuren über 255 Minuten

**Qualifikationsphase 2.2:** Vorabiklausur und Abiturarbeit

### **2.3.2 Beurteilungsbereich »Sonstige Mitarbeit«**

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

#### **Verbindliche Absprachen der Fachkonferenz**

- In der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen ein anwendungsbezogenes Softwareprodukt. Dies damit erbrachte Leistung wird in die Note für die sonstige Mitarbeit einbezogen.

#### **Leistungsaspekte**

##### **Mündliche Leistungen**

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

##### **Praktische Leistungen durch zielführende Arbeit mit Informatiksystemen**

- Implementierung, Test und Anwendung von Informatiksystemen

##### **Sonstige schriftliche Leistungen**

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen (LZKs)

In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, ist die Durchführung von mindestens einer schriftliche Übung pro Kurshalbjahr angestrebt, in anderen Kursen entscheidet die Lehrkraft über die Durchführung.

Schriftliche Übungen dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–9 Stunden.

- Bearbeitung von schriftlichen Aufgaben im Unterricht

## **Kriterien**

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,

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

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

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

Leistungsrückmeldungen können erfolgen

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

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

**Exkursionen** Interessierte Schülerinnen und Schüler können im Laufe der Qualifikationsphase an einer Exkursion zur FH Bonn-Rhein-Sieg teilnehmen.

### **4 Qualitätssicherung und Evaluation**

Der schulinterne Lehrplan wird regelmäßig überarbeitet. Der aktuelle Stand ist von Februar 2022.