

# **Viktoriaschule Aachen**

## **Curriculum Informatik**

Fassung Mai 2015

# Inhaltsverzeichnis

1 Fachgruppe Informatik .....	3
2 Schulinterner Lehrplan Sekundarstufe I .....	4
2.1 Vorbemerkungen .....	4
2.2 WP II in Stufe 8 .....	5
2.3 WP II in Stufe 9 .....	5
3 Schulinterner Lehrplan Sekundarstufe II .....	6
3.1 Vorbemerkung .....	6
3.2 Einführungsphase .....	7
3.2.1 Übersicht Unterrichtsvorhaben .....	7
3.2.2 Konkretisierte Unterrichtsvorhaben EF .....	8
3.3 Qualifikationsphase .....	19
3.3.1 Übersicht Unterrichtsvorhaben Q1 .....	19
3.3.2 Übersicht Unterrichtsvorhaben Q2 .....	21
3.3.3 Konkretisierte Unterrichtsvorhaben Q1 und Q2 .....	22
4 Leistungsbewertung .....	22
4.1 Bewertung von Kursarbeiten in der Sekundarstufe I .....	22
4.2 Bewertung von Klausuren in der Sekundarstufe II .....	22
4.2.1 Verbindliche Absprachen .....	22
4.2.2 Kriterien .....	23
4.2.3 Grundsätze der Leistungsrückmeldung und Beratung .....	23
4.3 Bewertung von Facharbeiten und besonderen Lernleistungen .....	23
4.4 Bewertung der sonstigen Mitarbeit in der Sekundarstufe II .....	24
4.4.1 Verbindliche Absprachen der Fachkonferenz .....	24
4.4.2 Leistungsaspekte .....	24
4.4.3 Kriterien .....	25
5 Arbeitsgemeinschaften .....	25
6 Qualitätssicherung und Evaluation .....	26
7 Anhang .....	26
7.1 Beispiele für Kursarbeiten und Klausuren .....	26
7.2 Beurteilungsbogen zur Facharbeit .....	26

# 1 Fachgruppe Informatik

Die Viktoriaschule ist ein Gymnasium in Trägerschaft der Evangelischen Kirche im Rheinland. Sie liegt in der Aachener Innenstadt nahe dem Hauptbahnhof. Der Einzugsbereich der Schülerschaft umfasst das gesamte Stadtgebiet und die angrenzenden Gemeinden der Städteregion Aachen. Eine Konsequenz dieses Umstandes ist, dass ein erheblicher Teil der Schüler nicht die Möglichkeit hat, außerhalb der Schule mit anderen ihrer Stufe Lerngemeinschaften zu bilden und gemeinsam an Hausaufgaben oder Projekten zu arbeiten.

In der Sekundarstufe I gehört ein Informatik-Kurs des WP II (Stufen 8 und 9) zum durchgängigen Angebot der Schule. Im Wahlverhalten der Schüler der Stufe 7 ist die Nachfrage für diesen Kurs gut, so dass die maximal 20 Plätze meistens ausgeschöpft werden. Unbefriedigend ist bisher der Anteil der Mädchen in den Kursen, der meist unter 20% liegt.

Aufbauend auf der dreizügigen Sekundarstufe I kommt in der Einführungsphase der Sekundarstufe II in der Regel ein Grundkurs Informatik zustande, der bis zum Abitur Bestand hat. Zu diesen Kursen gehören sowohl Schüler, die Informatik im WP II belegt hatten, als auch solche, die noch keine Erfahrung mit Inhalten der Informatik oder dem Programmieren gemacht haben. Die Bildung eines Leistungskurses ist wegen der Größe der Jahrgänge und wegen des Wahlverhaltens der Schüler sehr unwahrscheinlich und daher bisher nicht in Betracht gekommen.

Aufgrund des besonderen Oberstufenmodells der Viktoriaschule mit drei Leistungskursen, zu denen das Fach Informatik aus genannten Gründen nicht gehört, können die Schüler Informatik nicht als 3. Abiturfach wählen, wohl aber als 4. mündliches Fach. Insofern ist das Fach Informatik prinzipiell nicht vom Zentralabitur betroffen.

Das Angebot an Arbeitsgemeinschaften im Bereich der Informatik befindet sich noch im Aufbau. Seit 2013 gibt es eine AG für LEGO-Mindstorms als Initiative von Schülern. Zur Zeit läuft der Versuch einer Programmier-AG für die Unterstufe auf der Basis von Scratch. Geplant ist ein Informatik-Halbjahr im Rahmen des zweijährigen Rhythmus einer MINT-AG für die Mittelstufe. Dieses AG-Angebot ist von Bedeutung, um die Kenntnisse der Informationstechnologie unter den Schülern breiter zu streuen und insbesondere die Schülerinnen besser auf ihre Fächerwahl für WP II und die Oberstufe vorzubereiten.

Die Ausstattung der Viktoriaschule für den Informatik-Unterricht ist mit zwei Computerräumen, die über 18 bzw. 20 Computerarbeitsplätze verfügen, gut. Bei den im Unterricht eingesetzten Programmen legen wir Wert auf frei verfügbare und möglichst quelloffene Werkzeuge, die den Schülern auch außerhalb des Unterrichts auf verschiedenen Betriebssystemen zur Verfügung stehen. Insbesondere gehören dazu LibreOffice, Firefox, Filius, Python, Phase5, jEdit, Greenfoot, BlueJ, und Eclipse. Als internetbasierte Lernplattform führen wir zur Zeit das System Moodle ein.

Ziele der Fachgruppe für die weitere Entwicklung des Lehrplans sind die stärkere Nutzung der Moodle-Plattform für Materialien und Schüleraktivitäten, Fortbildungen, Zusammenarbeit mit InfoSphere der RWTH, Ausbau der AGs, ein Konzept für Schnupperstunden in Stufe 7 und die Berücksichtigung geschlechtsspezifischer Aspekte.

## 2 Schulinterner Lehrplan Sekundarstufe I

### 2.1 Vorbemerkungen

Informatik als reguläres Unterrichtsfach gibt es in Nordrhein-Westfalen für die Sekundarstufe I nur im Wahlpflichtbereich II für die Stufen 8 und 9.

Der Lehrplan für diesen Bereich stammt aus dem Jahr 1993 und ist damit veraltet. Er berücksichtigt weder das Internet bzw. World-Wide-Web in angemessener Form noch die Verbreitung mobiler IT und deren tägliche Handhabung durch die Schüler. Die Berücksichtigung der Inhalte dieses Lehrplans ist daher schwierig. Hinsichtlich der Ziele fehlt dem Lehrplan die Kompetenz-Orientierung. Im Lehrplan-Navigator des Ministeriums für Sekundarstufe I erscheint das Fach Informatik nicht.

Dementsprechend fehlen in NRW zugelassene Lehrwerke für das Fach Informatik. Ein Lehrwerk kann deshalb nicht auf dieselbe Weise wie in anderen Fächern die Grundlage für das schulinterne Curriculum bilden.

Nicht nur die Inhalte, sondern auch die Methoden der Informatik unterliegen einem stetigen Wandel. Die Lebenswelt der Schülerinnen und Schüler verändert sich hinsichtlich ihres Umgangs mit IT-Medien schon innerhalb weniger Jahre stark. Man betrachte hier zum Beispiel die Entwicklung des „Web 2.0“ bezüglich sozialer Netzwerke und aktiver Mitgestaltung der Online-Welt. Je konkreter ein Curriculum also Inhalte und Methoden festlegt, desto schneller veraltet es auch.

Das 60-Minuten-Modell der Viktoriaschule hat zur Folge, dass die WP-II-Fächer mit zweimal 60 Minuten pro Woche unterrichtet werden, also 15 Minuten weniger als im herkömmlichen 45-Minuten-Modell. Bezogen auf die beiden Schuljahre bedeutet das eine Reduzierung um ca. 20 Zeitstunden oder etwa acht Unterrichtswochen.

Der Zeitpunkt der Kursarbeiten wird durch die Mittelstufenkoordination zentral für alle WP-II-Kurse festgelegt und lässt sich daher nicht an didaktische Erfordernisse anpassen. Die Abstände zwischen den Arbeiten und gegenüber dem jeweiligen Beginn der Halbjahre schwanken stark und ändern sich auch von Schuljahr zu Schuljahr. Die Anzahl der Unterrichtsstunden und die Inhalte für die Themenreihen müssen daher immer wieder angepasst werden, damit zum jeweiligen Termin der Arbeiten ein sinnvoller Einschnitt erfolgen kann. Fakultative Inhalte sind in der Tabelle grau unterlegt.

In der Information für Eltern zur Sekundarstufe I in NRW von 2004 heißt es: „In den Klassen 7 bis 9 mit dem Schwerpunkt in der Klasse 8 wird eine informations- und kommunikationstechnologische Grundbildung vermittelt.“ (S. 20) Dementsprechend sollte das Fach Informatik nicht die Aufgabe übernehmen, in allgemeine Anwendungsprogramme (Office, Dateiverwaltung, Internet) einzuführen.

## 2.2 WP II in Stufe 8

Inhaltliche Schwerpunkte	Unterrichtsvorhaben, methodische Schwerpunkte und Kompetenzen
<b>Grundlagen digitaler Welten</b> - binäres Zahlensystem und elektronische Schalter - Text, Bild, Ton u.a. in digitaler Form - einfaches Rechnen mit Dualzahlen	Im Rahmen einer animierten Präsentation stellen die Schüler Aspekte der digitalen Welt dar.
<b>HTML als Beschreibungssprache des World-Wide-Web</b> - Geschichte und Struktur des WWW - Grundgerüst und allgemeine Objekte einer Webseite mit HTML - Hyperlinks, URLs und Bilder in Webseiten	Die Schüler gestalten Webseiten mit strukturierten Inhalten zu einem Oberthema und verknüpfen sie untereinander. Das Projekt wird als ganzes auf einen Webserver übertragen.
<b>Gestalten mit Stylesheets und Interaktionen (CSS, Grafik, Javascript)</b> - Farben und Box-Modell für Objekte einer Webseite - zentrale Formate für eine Auswahl an Objekten - flexible und vielgestaltige Layouts - Bearbeitung von Bildern - einfache Effekte mit JavaScript	Die zuvor erstellten Webseiten erhalten ein neues sowie vielfältigeres Design und Layout.
<b>Netzwerktechnik und Datenbanken</b> - Komponenten eines einfachen Netzwerks - Protokolle und Pakete im Internet - Struktur verbundener Netzwerke und Routing - Sicherheit in Netzwerken	Die Schüler experimentieren mit Netzwerk-Komponenten in der Simulationsumgebung Filius.

## 2.3 WP II in Stufe 9

Inhaltliche Schwerpunkte	Unterrichtsvorhaben, methodische Schwerpunkte und Kompetenzen
<b>Tabellenkalkulation als Einstieg ins Programmieren</b> - Datentypen und Berechnungen - Funktionen für Verweise, Texte und Suche - Programmieren mit Bedingungen	Die Schüler erstellen Tabellen für komplexe Berechnungen, z.B. BMI mit mehreren Parametern, Energiebilanz, Verschlüsselung geheimer Botschaften oder ein Spiel wie Mastermind.
<b>Grundschrte des Programmierens mit Python</b> - Rechnen in der interaktiven Shell (Interpreter) - Objekte und ihre Funktionen aufrufen - Wiederholte und bedingte Schritte - Variablen als Merkzettel - Ein- und Ausgabe mit Tastatur und Maus	Mit Python-Turtle, einem einfachen grafischen Werkzeugkasten, erstellen die Schüler Schritt für Schritt komplexere Grafiken und schließlich einfache Spiele mit kleinen Interaktionen.

Inhaltliche Schwerpunkte	Unterrichtsvorhaben, methodische Schwerpunkte und Kompetenzen
<p><b>Programmierprojekt mit Python für LEGO-Mindstorms (evtl. Raspberry-Pi)</b></p> <ul style="list-style-type: none"> <li>- Motoren, Sensoren und Aufbau des Roboter-Bausatzes</li> <li>- Komponenten des Roboters über Python-Funktionen steuern und abfragen</li> <li>- Parallele Prozesse für gleichzeitige Bewegungen von Motoren und Beobachtungen an Sensoren</li> <li>- Möglichkeiten und Grenzen von Robotern und künstlicher Intelligenz</li> </ul>	<p>Nach ersten Versuchen mit vorgegebenem Aufbau und einfachen Steueraufgaben haben die Schüler Gelegenheit, ein eigenes kleines Forschungsprojekt durchzuführen, bei dem der Roboter eine Aufgabe löst.</p> <p>Mögliche Kontexte:</p> <ul style="list-style-type: none"> <li>- selbstlenkende Fahrzeuge im Verkehr</li> <li>- Umgebungen erkunden und Strukturen erkennen</li> </ul>

## 3 Schulinterner Lehrplan Sekundarstufe II

### 3.1 Vorbemerkung

Die im Folgenden dargestellten Unterrichtsvorhaben decken die im Kernlehrplan angeführten Kompetenzen so ab, dass Lerngelegenheiten zur Erfüllung aller Kompetenzerwartungen gegeben sind.

Die Übersichtstabelle stellt die Verteilung der für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindlichen Unterrichtsvorhaben dar. Der ausgewiesene Zeitbedarf (bezogen auf Einheiten von 60 Minuten) 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, umfassen die Vorhaben dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, sprechen einige Unterrichtsvorhaben jeweils mehrere Inhaltsfelder an.

Für die Einführungsphase legen wir gemäß Beschluss der Fachkonferenz in weiten Strecken das Lehrwerk Informatik 1 aus dem Schöningh-Verlag zugrunde.

## 3.2 Einführungsphase

### 3.2.1 Übersicht Unterrichtsvorhaben

<p><u>Unterrichtsvorhaben E-I</u></p> <p><b>Thema:</b> Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Kommunizieren und Kooperieren</li><li>• Darstellen und Interpretieren</li><li>• Argumentieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Informatiksysteme</li><li>• Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Einsatz, Nutzung und Aufbau von Informatiksystemen</li><li>• Wirkung der Automatisierung</li></ul> <p><b>Zeitbedarf: 5 Stunden</b></p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p><b>Thema:</b> Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Formale Sprachen und Automaten</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Syntax und Semantik einer Programmiersprache</li></ul> <p><b>Zeitbedarf: 6 Stunden</b></p>
<p><u>Unterrichtsvorhaben E-III</u></p> <p><b>Thema:</b> Algorithmische Grundstrukturen in Java</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Argumentieren</li><li>• Modellieren</li><li>• Implementieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li></ul> <p><b>Zeitbedarf: 13 Stunden</b></p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p><b>Thema:</b> Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Kommunizieren und Kooperieren</li><li>• Darstellen und Interpretieren</li><li>• Argumentieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Informatiksysteme</li><li>• Informatik, Mensch und Gesellschaft</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Binäre Codierung und Verarbeitung</li><li>• Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten</li></ul> <p><b>Zeitbedarf: 6 Stunden</b></p>

<p><u>Unterrichtsvorhaben E-V</u></p> <p><b>Thema:</b> Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Kommunizieren und Kooperieren</li> <li>• Darstellen und Interpretieren</li> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Implementieren</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> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Objekte und Klassen</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p><b>Zeitbedarf: 14 Stunden</b></p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p><b>Thema:</b> Such- und Sortieralgorithmen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Daten und ihre Strukturierung</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen zum Suchen und Sortieren</li> <li>• Analyse, Entwurf und Implementierung einfacher Algorithmen</li> <li>• Objekte und Klassen</li> </ul> <p><b>Zeitbedarf: 7 Stunden</b></p>
<p><u>Unterrichtsvorhaben E-VII</u></p> <p><b>Thema:</b> Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Kommunizieren und Kooperieren</li> <li>• Darstellen und Interpretieren</li> <li>• Argumentieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Geschichte der automatischen Datenverarbeitung</li> <li>• Wirkungen der Automatisierung</li> <li>• Dateisystem</li> </ul> <p><b>Zeitbedarf: 8 Stunden</b></p>	

Gesamter Zeitbedarf für die Einführungsphase: 59 Unterrichtsstunden (a 60 Minuten)

### 3.2.2 Konkretisierte Unterrichtsvorhaben EF

#### *Didaktische Lernumgebung*

Zur Einführung in die objektorientierte Programmierung verwenden wir die didaktische Lernumgebung Greenfoot. Die Lehrtexte und Aufgaben des Lehrwerks „Informatik 1“ beziehen sich



auch speziell auf ein bestimmtes Greenfoot-Szenario.

Nach der Einarbeitungsphase sollte auf eine komplexere Lernumgebung umgestellt werden. Hier bieten sich verschiedene Möglichkeiten (in alphabetischer Reihenfolge) an:

- BlueJ
- Eclipse
- Java Editor
- NetBeans

Die folgenden Konkretisierungen von Vorhaben stammen aus einer Vorlage des Schöningh-Verlags und sind auf das eingeführte Lehrbuch „Informatik 1“ abgestimmt.

### ***Unterrichtsvorhaben EF-I***

**Thema:** Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik

**Leitfragen:** Was macht Informatik? Welche fundamentalen Konzepte müssen Informatikerinnen und Informatiker in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk- und Computersystem wiederfinden?

#### **Vorhabenbezogene Konkretisierung:**

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und Gesellschaft thematisiert. Das am Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

**Zeitbedarf:** 5 Stunden

#### **Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Informatiksysteme und ihr genereller Aufbau</b> (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)	Kapitel 1 „Was macht Informatik“ Als Anschauungsmaterial bieten sich Navigationsgeräte an
<b>2. Der kompetente Umgang mit dem Schulnetzwerk</b> (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		Kapitel 1 „Was macht Informatik“ Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule

### Unterrichtsvorhaben EF-II

**Thema:** Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

**Leitfragen:** *Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren?*

#### Vorhabenbezogene Konkretisierung:

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

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 4) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in Kapitel 2 zudem auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken.

Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben (Kapitel 3) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

**Zeitbedarf:** 6 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Identifikation von Objekten und Klassen</b></p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M),</li> <li>- stellen den Zustand eines Objekts dar (D),</li> <li>- modellieren Klassen mit ihren Attributen und ihren Methoden (M),</li> <li>- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>- implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).</li> </ul>	<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.1 Objektorientierte Modellierung</p>
<p><b>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</b></p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>		<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.2 Das Greenfoot-Szenario „Planetenerkundung“            Von der Realität zu Objekten            Von den Objekten zu Klassen, Klassendokumentation            Objekte inspizieren            Methoden aufrufen            Objektidentität und Objektzustand</p>
<p><b>3. Implementierung einfacher Aktionen in Greenfoot</b></p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p><b>Kapitel 2 Einführung in die Objektorientierung</b></p> <p>2.3 Programmierung in Greenfoot            Methoden schreiben            Programme übersetzen und testen</p>

**Unterrichtsvorhaben EF-III**

**Thema:** Algorithmische Grundstrukturen in Java

**Leitfragen:** *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

**Vorhabenbezogene Konkretisierung:**

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und

anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

**Zeitbedarf:** 13 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Algorithmen</b></p> <p>(a) Wiederholungen (While-Schleife)</p> <p>(b) bedingte Anweisungen</p> <p>(c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT</p> <p>(d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M),</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><b>Kapitel 3 Algorithmen</b></p> <p>3.1 Wiederholungen</p> <p>3.2 Bedingte Anweisungen</p> <p>3.3 Logische Funktionen</p> <p>3.4 Algorithmen entwickeln</p>
<p><b>2. Variablen und Methoden</b></p> <p>(a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife</p> <p>(b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert</p> <p>(c) Implementierung von Konstruktoren</p> <p>(d) Realisierung von Attributen</p>	<ul style="list-style-type: none"> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>- testen Programme schrittweise anhand von Beispielen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	<p><b>Kapitel 4 Variablen und Methoden</b></p> <p>4.1 lokale Variablen</p> <p>4.2 Methoden</p> <p>4.3 Attribute</p>

**Unterrichtsvorhaben EF-IV**

**Thema:** Das ist die digitale Welt! - Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

**Leitfragen:** Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten

zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der von-Neumann-Architektur erarbeitet werden.

**Zeitbedarf:** 8 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Analoge und digitale Aufbereitung und Verarbeitung von Daten</b></p> <p>(a) Erarbeitung der Unterschiede von analog und digital            (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A)</li> <li>- 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>	<p><b>Exkurs „Analog und Digital“</b></p>
<p><b>2. Der Umgang mit binärer Codierung von Informationen</b></p> <p>(a) Das binäre (und hexadezimale) Zahlensystem            (b) Binäre Informationsspeicherung            (c) Binäre Verschlüsselung            (d) Implementation eines Binärrechners</p>	<ul style="list-style-type: none"> <li>- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</li> </ul>	<p><b>Exkurs „Binäre Welt“</b></p>
<p><b>3. Aufbau informatischer Systeme</b></p> <p>(a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen            (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips</p>		<p><b>Exkurs „Arbeitsweise eines Computers“</b></p>

**Unterrichtsvorhaben EF-V**

**Thema:** Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

**Leitfragen:** Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf model-

liert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben hat die Entwicklung von Objekt- und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

**Zeitbedarf:** 18 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Umsetzung von Anforderungen in Entwurfsdiagramme</b></p> <p>(a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert</p> <p>(b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern eine objektorientierte Modellierung (A),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen,</li> </ul>	<p><b>Kapitel 6 Klassenentwurf</b></p> <p>6.1. Von der Realität zum Programm</p> <p>6.2. Objekte identifizieren</p> <p>6.3. Klassen und Beziehungen entwerfen</p>

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>2. Implementationsdiagramme als erster Schritt der Programmierung</b></p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden  (b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern  (c) Entwicklung von Klassendokumentationen  (d) Erstellung von Sequenzdiagrammen als Vorbereitung Vorbereitung für die Programmierung</p>	<p>Objekttypen oder lineare Datensammlungen zu (M),  - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),  - modellieren Klassen unter Verwendung von Vererbung (M),  - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),  - testen Programme schrittweise anhand von Beispielen (I),  - interpretieren Fehlermeldungen und korrigieren den Quellcode (I),  - analysieren und erläutern einfache Algorithmen und Programme (A)</p>	<p><b>Kapitel 6 Klassenentwurf</b>  6.4 Klassen und Beziehungen implementieren  6.5 Vererbung</p>
<p><b>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</b></p> <p>(a) Klassen werden in Java-Quellcode umgesetzt  (b) Das Geheimnisprinzip wird umgesetzt  (c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<p>- modifizieren einfache Algorithmen und Programme (I),  - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).  - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),  - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</p>	<p><b>Kapitel 6 Klassenentwurf</b>  6.4 Klassen und Beziehungen implementieren  6.5 Vererbung</p>
<p><b>4. Vererbungsbeziehungen</b></p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet  (b) Die Vorteile der Vererbungsbeziehungen  (c) Vererbung wird implementiert</p>		<p>Kapitel 6 Klassenentwurf  6.5 Vererbung</p>
<p><b>5. Softwareprojekt</b></p> <p>(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)  (b) Erweiterung des Spiels um weitere Funktionalitäten  (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche  (d) (arbeitsteilige) Implementation des Spiels</p>		<p><b>Kapitel 8 Softwareprojekte</b>  8.1 Softwareentwicklung  8.2 Oberflächen</p>

## Unterrichtsvorhaben EF-VI

**Thema:** Such- und Sortialgorithmen

**Leitfragen:** *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

**Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

**Zeitbedarf:** 9 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Modellierung und Implementation von Datenansammlungen</b></p> <p>(a) Modellierung von Attributen als Felder                      (b) Deklaration, Instanziierung und Zugriffe auf ein Feld</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)</li> <li>- entwerfen einen weiteren Algorithmus zum Sortieren (M)</li> <li>- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)</li> </ul>	<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>7.1 Das Feld – Eine Sammlung von Daten</p>
<p><b>2. Explorative Erarbeitung von Suchverfahren</b></p> <p>(a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion.                      (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.</p>	<ul style="list-style-type: none"> <li>- ordnen Attributen lineare Datenansammlungen zu (M)</li> </ul>	<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>Projekteinstieg 1: Suchen                      7.2 Suchen mit System                      Lineare Suche                      Binäre Suche                      Hashing</p>



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>(e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>Projekteinstieg 2: Sortieren</p> <p>7.3 Ordnung ist das halbe Leben!? – Sortieren</p> <p>Sortieren</p> <p>Selection Sort</p> <p>Insertion Sort</p> <p>Bubble Sort</p>

### *Unterrichtsvorhaben EF-VII*

**Thema:** Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

**Leitfragen:** Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Im ersten Schritt des Unterrichtsvorhabens wird anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

**Zeitbedarf:** 12 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Schriftzeichen, Rechenmaschine, Computer</b></p> <p>(a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden.</p> <p>(b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),</li> <li>- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A)</li> <li>- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> </ul>	<p>Exkurs „Geschichte der Informatik“</p>
<p><b>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</b></p> <p>(a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet</p> <p>(b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt</p> <p>(c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.</p>		<p>Exkurs „Informatik und Gesellschaft“</p>

## 3.3 Qualifikationsphase

### 3.3.1 Übersicht Unterrichtsvorhaben Q1

Für die folgende Übersicht gilt Analoges wie für die Übersicht zur EF (siehe 3.1.1).

Die Beschreibung stimmt zum großen Teil mit dem vom Ministerium bereitgestellten Beispiel und der zum entsprechenden Lehrwerk passenden Übersicht vom Schöningh-Verlag überein. Die Reihenfolge und einzelne Schwerpunkte hat die Fachkonferenz jedoch angepasst.

<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b> Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</p> <p><b>Zentrale Kompetenzen:</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>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</li><li>• Informatiksysteme</li></ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"><li>• Objekte und Klassen</li><li>• Analyse, Entwurf und Implementierung von Algorithmen</li><li>• Syntax und Semantik einer Programmiersprache</li><li>• Wirkung von Informatiksystemen</li></ul> <p><b>Zeitbedarf: 6 Stunden</b></p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b> Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Algorithmen</li><li>• Formale Sprachen und Automaten</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></ul> <p><b>Zeitbedarf: 15 Stunden</b></p>
---	--

<p><u>Unterrichtsvorhaben Q1-III</u></p> <p><b>Thema:</b> Suchen und Sortieren auf linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Modellieren</li> <li>• Implementieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Analyse, Entwurf und Implementierung von Algorithmen</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf: 12 Stunden</b></p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p><b>Thema:</b> Modellierung und Nutzung von ereignisgesteuerten grafischen Benutzeroberflächen in Anwendungskontexten</p> <p><b>Zentrale Kompetenzen:</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>• Informatiksysteme</li> <li>• Algorithmen</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Ereignisse verarbeiten</li> <li>• Vererbungs- und Objektstrukturen</li> <li>• Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf: 15 Stunden</b></p>
<p><u>Unterrichtsvorhaben Q1-V</u></p> <p><b>Thema:</b> Aufbau von und Kommunikation in Netzwerken</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Einzelrechner und Rechnernetzwerke</li> <li>• Sicherheit</li> <li>• Nutzung von Informatiksystemen, Wirkungen der Automatisierung</li> </ul> <p><b>Zeitbedarf: 7 Stunden</b></p>	

Gesamter Zeitbedarf für Q1: 55 Unterrichtsstunden (a 60 Minuten)

### 3.3.2 Übersicht Unterrichtsvorhaben Q2

<p><u>Unterrichtsvorhaben Q2-I</u></p> <p><b>Thema:</b> Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Darstellen und Interpretieren</li> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Formale Sprachen und Automaten</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> </ul> <p><b>Zeitbedarf: 16 Stunden</b></p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p><b>Thema:</b> Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten</p> <p><b>Zentrale Kompetenzen:</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>• Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Datenbanken</li> <li>• Algorithmen in ausgewählten informatischen Kontexten</li> <li>• Syntax und Semantik einer Programmiersprache</li> <li>• Sicherheit</li> </ul> <p><b>Zeitbedarf: 15 Stunden</b></p>
<p><u>Unterrichtsvorhaben Q2-III</u></p> <p><b>Thema:</b> Endliche Automaten und formale Sprachen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Argumentieren</li> <li>• Modellieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Endliche Automaten und formale Sprachen</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Endliche Automaten</li> <li>• Grammatiken regulärer Sprachen</li> <li>• Möglichkeiten und Grenzen von Automaten und formalen Sprachen</li> </ul> <p><b>Zeitbedarf: 12 Stunden</b></p>	

Gesamter Zeitbedarf für Q2: 43 Unterrichtsstunden

### **3.3.3 Konkretisierte Unterrichtsvorhaben Q1 und Q2**

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind.

Mögliche Vorhaben werden mit Einführung des noch nicht erschienen Lehrbuchs Informatik 2 aus dem Schöningh-Verlag ergänzt. Bis dahin nutzen die Lehrkräfte ihr bisher eingesetztes Material.

## **4 Leistungsbewertung**

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

### **4.1 Bewertung von Kursarbeiten in der Sekundarstufe I**

Pro Halbjahr schreiben die Schüler zwei Kursarbeiten von jeweils 60 Minuten Dauer. In Stufe 9 kann eine der schriftlichen Arbeiten durch eine Projektarbeit ersetzt werden.

Die Bewertung der schriftlichen Leistungen in den Kursarbeiten erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Teillösungen zugeordnet sind.

Die Note ausreichend soll bei Erreichen von 50% der Hilfspunkte erteilt werden.

### **4.2 Bewertung von Klausuren in der Sekundarstufe II**

#### **4.2.1 Verbindliche Absprachen**

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

Instrumente:

Einführungsphase: 1 Klausur je Halbjahr                      Dauer der Klausur: 90 Minuten

Grundkurse Q 1: 2 Klausuren je Halbjahr Dauer der Klausuren: 90 Minuten

Grundkurse Q 2.1: 2 Klausuren Dauer der Klausuren: 140 Minuten

In der Einführungsphase wird aufgrund nur einer Klausur der schriftliche Anteil etwa im Verhältnis 40:60 zur sonstigen Mitarbeit gewertet.

Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

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

## 4.2.2 Kriterien

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

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

## 4.2.3 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 (mit angemessener Bedenkzeit),
- 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 Grundkursfach in der Qualifikationsphase.

## 4.3 Bewertung von Facharbeiten und besonderen Lernleistungen

Grundlegende Aussagen zu Facharbeiten, Projektkursen und besonderen Lernleistungen enthält das allgemeine Konzept der Leistungsbewertung der Viktoriaschule. Darüber hinaus einigte sich die

Fachkonferenz Informatik auf ein mögliches Raster zur Bewertung der jeweils abzugebenden Arbeit (siehe Facharbeitsbeurteilungsbogen).

Bei Einbringen einer besonderen Lernleistung in die Berechnung der Abiturnote muss die zugrunde gelegte Arbeit einen deutlich größeren Umfang bzw. ein höheres Niveau aufweisen als eine Facharbeit.

## **4.4 Bewertung der sonstigen Mitarbeit in der Sekundarstufe II**

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

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

Alle Schülerinnen und Schüler führen in der Einführungsphase in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Arbeitsmappe mit Arbeitstagebuch an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

In der Qualifikationsphase erstellen, dokumentieren und präsentieren die Schülerinnen und Schüler in Kleingruppen ein anwendungsbezogenes Softwareprodukt. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

### **4.4.2 Leistungsaspekte**

Die Beiträge für den Unterricht lassen sich den folgenden drei Aspekten zuordnen. Für ihre Gewichtung soll das Verhältnis 40 : 40 : 20 als Orientierung dienen.

#### *Mündliche Leistungen*

- Beteiligung am Unterrichtsgespräch bzw. am Gespräch in der Gruppe
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate

#### *Praktische Leistungen am Computer*

- Implementierung und Test von Informatiksystemen
- Modellierung in Form von Diagrammen (UML)

#### *Sonstige schriftliche Leistungen*

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben (z.B. Projekt)
- Beiträge in Foren und Wiki-Seiten des Kurses
- Bearbeitung von schriftlichen Aufgaben im Unterricht
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen

Der Einsatz von Moodle als webbasierter Lernplattform bietet die Möglichkeit, dass die Schülerinnen und Schüler dort schriftliche Beiträge auf freiwilliger Basis, als Einreichung oder als Überprüfung leisten.

Über die Durchführung schriftlicher Übungen entscheidet die Lehrkraft nach Bedarf. Sie dauern ca.



20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.

### 4.4.3 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 Ziel- und Ergebnisorientierung,
- die angemessene Verwendung der Fachsprache und der Fachmethoden,
- die Anschaulichkeit durch Beispiele und symbolische Darstellungen,
- die Komplexität und den Grad der Abstraktion,
- und die Präzision zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Arbeitsteilung durch Modularisieren und Einhalten von Absprachen,
- die Durchführung fachlicher Arbeitsanteile,
- das Dokumentieren und Bereitstellen von Teilergebnissen und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

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

## 5 Arbeitsgemeinschaften

Wie im Kapitel 1 erwähnt, besteht mit der aufgrund einer Schülerinitiative 2013 begonnenen Lego-Mindstorms-AG ein Angebot im Bereich der Informatik. Der Schwerpunkt hat sich seither verlagert. Zunächst war besonders die Mittelstufe als Zielgruppe intendiert, allerdings stellte sich das Interesse in der Unterstufe als deutlich größer heraus. Die AG wird von zwei Schülern der Oberstufe betreut, wobei die Kontinuität davon abhängt, wie lange sie noch auf der Schule verbleiben und ob ihr Stundenplan zur AG-Zeit eine Lücke bietet. Programmiert wird mit der auf symbolischen Blöcken basierenden Software von Lego. Bei einigen Schülern besteht aber der Wunsch, eine höhere Programmiersprache einzusetzen (Java, NXC oder Python). Es ist erforderlich, dass ein Lehrer die AG bezüglich didaktischer und organisatorischer Aspekte betreut.

Als weiteres Angebot soll eine Programmier-AG für die Unterstufe dauerhaft eingerichtet werden. In dieser Initiative von Studenten der RWTH kommt als Programmierumgebung die webbasierte Version von Scratch zum Einsatz. Für die Kontinuität der Betreuung und die Begleitung durch einen Lehrer gilt ähnliches wie bei der Mindstorms-AG.

Die Fortführung und der Ausbau des AG-Angebots hängen wesentlich von der verfügbaren Zeit der

Informatiklehrer ab. Inwieweit die Schule diese Ressourcen gewährleisten kann, ist nicht sicher.

## 6 Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Projektphasen 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. Im Sommer 2016 werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtbewertung des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.

## 7 Anhang

### 7.1 Beispiele für Kursarbeiten und Klausuren

Werden später ergänzt.

### 7.2 Beurteilungsbogen zur Facharbeit

Grundsätze:

- Das Verhältnis Form und Inhalt in der Notengebung beträgt 50:50.
- Dieser Beurteilungsbogen ergänzt das schriftliche Gutachten.

#### Formale Leistung (50 Punkte)

Der Prüfling		max.	S.
1	legt eine vollständige Arbeit vor: korrektes Deckblatt, Inhaltsverzeichnis, Literaturverzeichnis, Erklärung.	4	
2	hält sich an den vereinbarten Umfang.	2	
3	hält typographische Vorgaben ein: Einband, Seitenspiegel, Seitenangaben, gliedernde Abschnitte und Überschriften, Schriftgrad.	10	
4	hält sich an seine Gliederung: Inhaltsverzeichnis und Textteil stimmen überein.	3	
5	ordnet der Darstellung Quellen zu.	5	
6	zeigt einen formal korrekten Umgang mit seinen Quellen.	4	
7	legt ein Literaturverzeichnis mit allen Angaben zu den benutzten Hilfsmitteln vor (Literatur, Abbildungen, Materialien) und speichert alle Internet-Quellen als Anhang.	4	

8	schreibt sprachlich richtig (Grammatik, Syntax, Orthografie, Zeichensetzung) und stilistisch sicher sowie variabel.	10	
9	stellt Diagramme nach UML-Standard und Quelltexte in besonderer Formatierung mit Zeilennummern dar.	8	
	<b>Summe</b>	<b>50</b>	

### **Inhaltliche und gedankliche Leistung (50 Punkte)**

#### **Der Prüfling**

1	wählt in Absprache mit dem Lehrer ein eigenständiges Thema und eine eigenständige Problemstellung.	4	
2	strukturiert seinen Text schlüssig, stringent sowie gedanklich klar und stellt durchgängig und konsequent einen Themenbezug her.	5	
3	nutzt genügend, sinnvolle und sachgemäße Quellen; recherchiert umfangreich.	5	
4	verwertet seine Quellen im Sinne des Themas und der Problemstellung: verknüpft seine Argumentation sinnvoll mit Nachweisen.	4	
5	argumentiert sinnvoll und nachvollziehbar, bezieht beschreibende, deutende und wertende Aussagen schlüssig aufeinander, unterscheidet gewissenhaft zwischen Faktendarstellung, fachwissenschaftlichen Positionen und der eigenen Meinung.	4	
6	formuliert unter Beachtung der Fachsprache sachlich, präzise und begrifflich differenziert	8	
7	kommt zu gehaltvollen Erarbeitungen sowie Teil- und Gesamtergebnissen. [Kommentar dazu: siehe Korrekturbemerkungen in der Arbeit und beiliegendes Gutachten]	20	
	<b>Summe</b>	<b>50</b>	

	<b>Summe der formalen und inhaltlichen Leistung</b>	<b>100</b>	
	<b>Note</b>		