

**Schulinterner Lehrplan  
zum Kernlehrplan für die gymnasiale Oberstufe  
*Einführungsphase***

# **Fach Informatik**

**Inhalt:**

- |   |         |
|---|---------|
| 1. Übersicht über die Inhaltsfelder und Kompetenzen | Seite 2 |
| 2. Entscheidungen zum Unterricht                    | Seite 4 |
| 2.1 Übersichtsraster Unterrichtsvorhaben            | Seite 5 |
| 2.2 Konkretisierte Unterrichtsvorhaben              | Seite 8 |

## 1. Übersicht über die Inhaltsfelder und Kompetenzen

### Übersicht über die Inhaltsfelder (allgemein)

- I. Daten und ihre Strukturierung
- II. Algorithmen
- III. Formale Sprachen und Automaten
- IV. Informatiksysteme
- V. Informatik, Mensch und Gesellschaft

### Übersicht der konkretisierten Kompetenzerwartungen (für 10EF)

#### Argumentieren (A)

Die Schülerinnen und Schüler

- A1. analysieren und erläutern eine objektorientierte Modellierung
- A2. beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf
- A3. analysieren und erläutern einfache Algorithmen und Programme
- A4. beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“
- A5. bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen
- A6. erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung

#### Modellieren (M)

Die Schülerinnen und Schüler

- M1. ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen
- M2. modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen
- M3. modellieren Klassen unter Verwendung von Vererbung
- M4. ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu
- M5. ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu
- M6. stellen die Kommunikation zwischen Objekten grafisch dar
- M7. entwerfen einen weiteren Algorithmus zum Sortieren
- M8. entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar

### **Implementieren (I)**

Die Schülerinnen und Schüler

- I1. implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken
- I2. implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen
- I3. testen Programme schrittweise anhand von Beispielen
- I4. modifizieren einfache Algorithmen und Programme
- I5. implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache
- I6. interpretieren Fehlermeldungen und korrigieren den Quellcode

### **Darstellen und Interpretieren (D)**

Die Schülerinnen und Schüler

- D1. stellen den Zustand eines Objekts dar
- D2. stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar
- D3. dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden
- D4. analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an
- D5. stellen ganze Zahlen und Zeichen in Binärcodes dar
- D6. interpretieren Binärcodes als Zahlen und Zeichen
- D7. nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst

### **Kommunizieren und Kooperieren (K)**

Schülerinnen und Schüler

- K1. nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung
- K2. nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation

## 2. Entscheidungen zum Unterricht

Die nachfolgend dargestellte Umsetzung der verbindlichen Kompetenzerwartungen des Kernlehrplans findet auf zwei Ebenen statt: der Übersichts- und der Konkretisierungsebene. Sie orientiert sich damit an den Vorschlägen des Schulministeriums.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den 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. 80 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.2) 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.

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

Laut Beschluss der Fachkonferenz erfolgt die Umsetzung der Anwendungskontexte in der **Programmiersprache Java** gemäß den Vorgaben des Schulministeriums. Als Standardentwicklungsumgebungen werden dabei BlueJ und Greenfoot verwendet. Die unterrichtende Lehrkraft kann jedoch weitere Entwicklungsumgebungen für Java kontextbezogen einzuführen und verwenden.

## 2.1 Übersichtsraster Unterrichtsvorhaben

Einführungsphase	
<p><b><u>Unterrichtsvorhaben E-I</u></b></p> <p><b>Thema:</b></p> <p><i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i></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</li> <li>• Dateisystem</li> <li>• Internet</li> <li>• Einsatz von Informatiksystemen</li> </ul> <p><b>Zeitbedarf:</b> ca. 8 Stunden</p>	<p><b><u>Unterrichtsvorhaben E-II</u></b></p> <p><b>Thema:</b></p> <p><i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Modellieren</li> <li>• Implementieren</li> <li>• Darstellen und Interpretieren</li> <li>• Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Formale Sprachen und Automaten</li> </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:</b> ca. 20 Stunden</p>

**Einführungsphase**

**Unterrichtsvorhaben E-III**

**Thema:**

*Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** ca. 18 Stunden

**Unterrichtsvorhaben E-IV**

**Thema:**

*Such- und Sortieralgorithmen anhand kontextbezogener Beispiele*

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

- Algorithmen

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** ca. 9 Stunden

Einführungsphase	
<p><b><u>Unterrichtsvorhaben E-V</u></b></p> <p><b>Thema:</b> <i>Vertiefen der objektorientierten Analyse, Modellierung und Implementierung an Hand eines eigenen Projektes</i></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>• Informatiksysteme</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:</b> ca. 15 Stunden</p>	<p><b><u>Unterrichtsvorhaben E-VI</u></b></p> <p><b>Thema:</b> <i>Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</i></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>• Informatik, Mensch und Gesellschaft</li> <li>• Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>• Wirkungen der Automatisierung</li> <li>• Geschichte der automatischen Datenverarbeitung</li> <li>• Digitalisierung</li> </ul> <p><b>Zeitbedarf:</b> ca. 10 Stunden</p>
<p><b><u>Summe Einführungsphase ca. 80 Stunden</u></b></p>	

## 2.2 Konkretisierte Unterrichtsvorhaben

**Hinweis:** Thema, Inhaltsfelder, inhaltliche Schwerpunkte und Kompetenzen hat die Fachkonferenz des Paul-Klee-Gymnasiums Overath verbindlich vereinbart. In allen anderen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen bei der Konkretisierung der Unterrichtsvorhaben möglich.

### Unterrichtsvorhaben EF-I

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

**Leitfragen:** *Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?*

#### **Vorhabenbezogene Konkretisierung:**

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

Zunächst wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Anschließend wird auf die Übertragung von Informationen bzw. Daten im Sinne des Sender-Empfänger-Modells eingegangen. Dabei wird eine überblickartige Vorstellung der Kommunikation von Rechnern in Netzwerken erarbeitet.

Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Bei der Beschäftigung mit Datenkodierung, Datenübermittlung und Datenverarbeitung ist jeweils ein Bezug zur konkreten Nutzung der informatischen Ausstattung der Schule herzustellen. So wird in die verantwortungsvolle Nutzung dieser Systeme eingeführt.

**Zeitbedarf:** *ca. 8 Stunden*



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Information, deren Kodierung und Speicherung</b></p> <ul style="list-style-type: none"> <li>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</li> <li>(b) Darstellung von Informationen in Schrift, Bild und Ton</li> <li>(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</li> <li>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)</li> </ul>	<p>Beispiele:</p> <ul style="list-style-type: none"> <li>• Zahlen in Binär- /Hexadezimalcode: Umrechnen von Zahlen in Binär- und Hexadezimalcode</li> <li>• Textkodierung: Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Windings oder Ascii-Tabelle)</li> <li>• Bildkodierung: Kodierung von Bildinformationen in Raster- und Vektorgrafiken (RGB)</li> </ul>
<p><b>2. Informations- und Datenübermittlung in Netzen</b></p> <ul style="list-style-type: none"> <li>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</li> <li>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server.</li> <li>(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</li> <li>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).</li> <li>• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).</li> </ul>	<p>Beispiele:</p> <ul style="list-style-type: none"> <li>• Das OSI-Schichtenmodell: Die Schülerinnen und Schüler recherchieren im Internet das OSI-Schichtenmodell und erfahren z. B. an Hand einer E-Mail wie Anwendungen auf verschiedenen Architekturen Daten tauschen.</li> <li>• Das TCP/IP-Protokoll</li> <li>• Routing und Internetdomain: Rollenspiel zur Paketvermittlung und Ausfall von Knoten</li> <li>• Aufbau des Schulnetzwerks: Die erlernten Modelle zum Informationsaustausch werden im Schulnetzwerk identifiziert.</li> </ul>

**3. Aufbau informatischer Systeme**

- (a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“
- (b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“

Die Schülerinnen und Schüler

- beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).

Material: Papierrechner, Demonstrationshardware

- Mit dem Papierrechner führen die SchülerInnen einfache Rechnungen durch, erarbeiten damit die von-Neumann-Architektur“ und identifizieren diese in einem ausrangierten Rechner

## **Unterrichtsvorhaben EF-II**

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

**Leitfrage:** *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation informatisch realisieren?*

### **Vorhabenbezogene Konkretisierung:**

Ein zentraler Bestandteil des Informatikunterrichts der EF ist die objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementation an Hand von einfachen Beispielen und Klassen ein.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des Objektorientierten Paradigma strukturiert. Es werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektarten und Klassendiagramme eingeführt.

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

**Zeitbedarf:** ca. 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Identifikation von Objekten</b></p> <ol style="list-style-type: none"> <li>1. Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt</li> <li>2. Objekte werden mit Klassendiagrammen visualisiert und mit sinnvollen Attributen und „Fähigkeiten“ d.h. Methoden versehen.</li> <li>3. Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Klasse zusammengefasst</li> <li>4. Vertiefung: Modellierung weiter Beispiele ähnlichen Musters</li> </ol>	<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>• modellieren Klassen mit ihren Attributen und Methoden (M)</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I)</li> <li>• stellen Abläufe und Programme grafisch dar (D,M).</li> </ul>	<p>Beispiele:</p> <ul style="list-style-type: none"> <li>• Aufzug (Klassendiagramm) mit Implementation der Attribute und einfacher Methoden (get/set-Methoden) in BlueJ</li> <li>• Auto: Die Schülerinnen und Schüler erstellen ein Klassendiagramm, definieren Attribute und Methoden und implementieren diese in BlueJ.</li> </ul>
<p><b>2. Implementation einfacher Klassen</b></p> <ol style="list-style-type: none"> <li>1. Grundaufbau einer Java Klasse</li> <li>2. Deklaration und Initialisierung von Objekten</li> <li>3. Konstruktoren</li> <li>4. einfache Methodenaufrufe</li> <li>5. Dokumentation einer Klasse</li> </ol>		
<p><b>3. Analyse von Klassen</b> Teilanalyse der Klasse</p>	<p>Die Schülerinnen und Schüler stellen Attribute und Methoden in einem Klassendiagramm dar. (D)</p>	<p>Beispiel:</p> <ul style="list-style-type: none"> <li>• Roboterszenario (Darstellung der gegebenen Klasse als Klassendiagramm)</li> </ul>

**4. Bewegungsanimation oder Ablaufsimulation**

1. kontinuierliches Bewegen oder Veränderung eines Objekts / Zustands mit Hilfe einer Schleife (WHILE-Schleife)
2. Bedingte Anweisungen (IF-Anweisungen)
3. Verändern eines Objektes mit Hilfe von Zählschleifen (FOR-Schleife)
4. Abfragen von Zuständen

Die Schülerinnen und Schüler

- analysieren und erläutern einfache Algorithmen und Programme (A),
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),
- modellieren Klassen mit Attributen und Methoden (M)
- ordnen Attributen, Parameter und Rückgaben von Methoden einfache Datentypen zu, (M)
- modifizieren einfache Algorithmen und Programme (I)
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- interpretieren Fehlermeldungen und testen Programme schrittweise (I).

Beispiele:

- Ampel: Die Schülerinnen und Schüler erstellen ein Klassendiagramm zu einer Ampel und simulieren den Schaltvorgang. Sie testen die Klasse mit einer vorgegebenen GUI.
- Roboterszenario (Hindernisse, Gegenstände aufsammeln/ablegen, suchen, IF-Abfragen zur Erkennung von Hindernissen und Gegenständen)
- Turtle-Grafik: Die Schülerinnen und Schüler zeichnen von Figuren mit Hilfe von Schleifen, Methodenaufrufe mit Parameter z.B. n-Eck, Fontänen, Haus vom Nikolaus...)

### **Unterrichtsvorhaben EF-III**

Thema: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten

**Leitfragen:** *Wie lassen sich komplexere Szenarien mit mehreren Objekten realisieren? Wie sind die Datenflüsse zwischen den Objekten und die Beziehungen zwischen den zugehörigen Klassen?*

#### **Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau von Objektbeziehungen und ihrer Realisation. Während im vorangegangenen Unterrichtsvorhaben der Aufbau und die Realisation einer Klasse im Vordergrund stehen, werden hier komplexere Kontexte untersucht, in denen mehrer Objekte untereinander Daten tauschen. Die Schülerinnen und Schüler müssen hierfür mehrere Klassen erstellen und diese in einer Szene als Ganzes darstellen. Es wird ein Implementationsdiagramm mit Objektreferenzen und Assoziationsbeziehungen der Klassen eingeführt.

Der Aufbau der Implementationsdiagramme erfolgt dabei nach und nach, in dem zunächst vorgefertigte Java-Klassen (wie die Klasse Math) in einer neuen Klasse eingebunden werden und danach komplexe Szenarien entwickelt werden, in denen eigene zu erstellende Objekte eingebunden werden müssen. Dazu bedarf es zunächst einer präzisen Unterscheidung zwischen Objektreferenzen und Objekten, so dass klar wird, dass Dienste eines Objektes von unterschiedlichen Objekten über unterschiedliche Referenzen in Anspruch genommen werden können. Es werden hier Szenarien verwendet, in denen mehrer Instanzen einer Klasse erzeugt werden. Dabei wird die Komplexität stetig gesteigert und schließlich ein Projekt realisiert, in denen die Instanzen in einem Array gespeichert werden. Es kann hierbei auf das Prinzip der abstrakten Klasse eingegangen werden, aber der Inhalt ist nicht obligatorisch. Auch das Prinzip der Vererbung im objektorientierten Sinne wird angesprochen und anhand spielerischer Beispiele geplant und realisiert.

#### **Inhaltliche Schwerpunkte:**

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

#### **Inhaltsfelder:**

*Algorithmen, Daten und ihre Strukturierung, Formale Sprachen und Automaten*

**Zeitbedarf:** ca. 18 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>Einführung von Referenztypen, Assoziationen und des Implementationsdiagramm</p> <p>a) Einfache Objektreferenzen</p> <ul style="list-style-type: none"> <li>- Einbinden von speziellen Referenzklassen in einer Klasse (String, Math, etc)</li> <li>- Verwenden der Klasse als Referenzklasse</li> <li>- Darstellung im einfachen Implementationsdiagramms</li> </ul> <p>b) Vertiefung der Objektrefrenzen und Einführung in Assoziationen mit Textobjekten</p> <ul style="list-style-type: none"> <li>- Verwenden der Stringoperatoren</li> <li>- Speichern und öffnen von Textdateien</li> <li>- Darstellung von Assoziationen</li> </ul> <p>c) Vertiefung mit weiteren Beispielen</p> <ul style="list-style-type: none"> <li>- Erstellen der Klassen- und des Implementationsdiagramms</li> <li>- Implementation der Grundklassen</li> <li>- Implementation der Objektreferenzen und Assoziationen</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M)</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M)</li> <li>• analysieren und erläutern eine objektorientierte Modellierung (A)</li> <li>• analysieren und erläutern einfache Algorithmen und Programme (A)</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> </ul>	<p>Beispiele</p> <ul style="list-style-type: none"> <li>• Bruchrechner: Ein Klasse Bruch wird mit der Referenzklasse Math erstellt, die die üblichen Bruchrechnungen durchführt. Dann gibt der Lehrer eine GUI vor, in der die Bruchklasse als Referenzklasse verwendet wird.</li> <li>• Krümel: Das Krümelmonster hat Kekse und einen Würfelbecher, in dem drei Würfel sind. Es wird ein Implemtationsdiagramm erstellt mit drei Klassen Krümel, Würfelbecher und Würfel. Abhängig vom Wurf Ergebnis erhält oder verliert Krümel Kekse.</li> <li>• Hangman: Das gleichnamige Spiel wird realisiert unter der Verwendung der Referenzklasse String. Dabei recherchieren die Schülerinnen und Schüler selbstständige die nötigen Methoden der Klasse String.</li> <li>• Wortsalat: Gespeicherte Wörter werden zufällig gezogen und die Buchstaben in beliebiger Reihenfolge dargestellt. Das dargestellte Wort muss erraten werden</li> </ul>



<p>Projektplanung unter Verwendung mehrerer Klassen mit Hilfe von Entwurfs- und Implementationsdiagrammen an komplexerem Beispiel mit mehreren Objektreferenzen</p> <ol style="list-style-type: none"> <li>Benenne der benötigten Objekte</li> <li>Erstellen der Klassendiagramme</li> <li>Aufstellen der Assoziationen</li> <li>Erstellen des Implementationsdiagramms</li> <li>Anpassen der Klassendiagramme und Benennung der Referenztypen</li> </ol>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> <li>modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M)</li> <li>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M)</li> <li>stellen die Kommunikation zwischen Objekten grafisch dar (M)</li> <li>stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)</li> </ul>	<p>Beispiel:</p> <ul style="list-style-type: none"> <li>Adressbuch: Die Schülerinnen und Schüler müssen ein Adressbuch mit einer bestimmten Anzahl von Einträgen realisieren. Dazu müssen die Einträge, die Verwaltung der Einträge und die Speicherung als Klasse definiert werden und in einem Implementationsdiagramm dargestellt werden.</li> </ul>
<p>Implementierung des Projekts</p> <ol style="list-style-type: none"> <li>Einführung von Arrays und deren Verwendung</li> <li>Speichern von Klassen in Array</li> <li>Erstellen der Referenzklassen</li> <li>Implementation der Referenzklassen und der Assoziationen</li> </ol>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> <li>modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M)</li> <li>ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M)</li> </ul>	<p>Beispiel:</p> <ul style="list-style-type: none"> <li>Adressbuch: Die Schülerinnen und Schüler müssen ein Adressbuch mit einer bestimmten Anzahl von Einträgen realisieren. Dazu müssen die Einträge, die Verwaltung der Einträge und die Speicherung als Klasse definiert werden und in einem Implementationsdiagramm dargestellt werden.</li> </ul>



	<ul style="list-style-type: none"> <li>• analysieren und erläutern eine objektorientierte Modellierung (A)</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</li> <li>• stellen den Zustand eines Objekts dar(D)</li> </ul>	
<p>Entwicklung einer komplexeren Simulation mit Objekten, die von einem Grundobjekt abgeleitet werden (Vererbung mit Ergänzung und Überschreiben der Methoden)</p> <ol style="list-style-type: none"> <li>Analyse und Erläuterung einer gegebenen Klasse eines Grundobjektes</li> <li>Spezialisierung der Klasse zu einer Unterklassen mit neuen Methoden</li> <li>Spezialisierung der Klasse zu einer Unterklasse durch Überschreiben einer Methode</li> <li>Darstellung der Klasse in Klassendiagramm</li> <li>Vertiefung durch Entwicklung eines weiteren Projektes</li> </ol>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• modellieren Klassen unter Verwendung von Vererbung (M)</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M)</li> <li>• analysieren und erläutern eine objektorientierte Modellierung (A)</li> <li>• analysieren und erläutern einfache Algorithmen und Programme (A)</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> </ul>	<p>Beispiele</p> <ul style="list-style-type: none"> <li>• Schatzräuber: Die Schülerinnen und Schüler erhalten eine Greenfoot-Vorlage mit einem Spielplan und einer Spielfigur. Sie analysieren die Klassen. Aus der Klasse der Spielfigur entwickeln sie einen Schatzräuber und einen Wächter, die neben den Bewegungen weitere Methoden haben.</li> <li>• Jäger und Beute: In Greenfoot wird eine Landschaft mit zwei Tierarten simuliert. Dabei haben die Tierarten, die gleichen Grundmethoden, die in einer Klasse festgelegt sind. Durch Vererbung werden aus der Klasse zwei Unterklassen gebildet, die Beute und die Jäger. Beide Tierarten sterben unter bestimmten Bedingungen aus. Durch Veränderung der Startsituation wird ein Gleichgewicht hergestellt.</li> </ul>

## **Unterrichtsvorhaben EF-IV**

**Thema:** Such- und Sortialgorithmen anhand kontextbezogener Beispiele

**Leitfragen:** *Wie wird in linearen Datenstrukturen effizient gesucht und sortiert?*

### **Vorhabenbezogene Konkretisierung:**

Als Vorbereitung auf die komplexeren Sortialgorithmen in der QI wird hier mit Hilfe von Arrays das Sortieren in Form von einfachen Algorithmen realisiert. Im Umgang mit Strings wird die Suche von Textteilen thematisiert. Gleichzeitig wird die vorangegangene Unterrichtseinheit vertieft durch das Verwenden von der Klasse String und Arrays. Die erstellten Projekte können hierfür verwendet und erweitert werden.

**Inhaltliche Schwerpunkte:** Analyse, Entwurf und Implementierung von Minsort, Maxsort, Bubblesort, wahlweise auch mit Bucketsort und Insertionsort, sowie die Beurteilung der jeweiligen Effizienz.

### **Inhaltsfelder:**

*Algorithmen, Daten und ihre Strukturierung, Formale Sprachen*

**Zeitbedarf:** *ca. 9 Stunden*

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>Erarbeitung und Darstellung eines Sortieralgorithmus anhand eines praktischen Beispiels</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• entwickeln einen Sortieralgorithmus (M)</li> <li>• stellen ihr Ergebnis im Plenum vor (D, K)</li> </ul>	<p>Beispiel:</p> <ul style="list-style-type: none"> <li>• Simulation einer Sortierung an Hand eines Kartenspiel, nummerierter Klebezettel oder einer Menschengruppe</li> </ul>
<p>Einführung konkreter Sortierverfahren wie z.B. Min- und Maxsort, Bubblesort, Bucketsort, Insertionsort.</p> <ol style="list-style-type: none"> <li>a) Erarbeitung der Verfahren in arbeitsteiliger Gruppenarbeit</li> <li>b) Vergleich der Algorithmen bezüglich ihrer Effizienz</li> <li>c) Implementierung</li> <li>d) Steigerung der Effizienz der Algorithmen</li> </ol>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• nutzen die erworbenen informatischen Kenntnisse zur Implementierung (A)</li> <li>• reflektieren kommunizieren ihre Ergebnisse im Vergleich der Gruppenergebnisse (K, D)</li> </ul>	<p>Beispiele:</p> <ul style="list-style-type: none"> <li>• Zahlenliste: Der Lehrer stellt eine Vorlage mit einer GUI zur Verfügung, in der Zahlen in einem Array gespeichert werden können. Die Schülerinnen und Schüler programmieren die Sortierverfahren und testen diese mit Hilfe der GUI.</li> <li>• Adressbuch: Das erstellte Adressbuch wird erweitert, so dass die Einträge gesucht und sortiert werden können.</li> </ul>

## **Unterrichtsvorhaben EF-V**

Thema: Vertiefen der objektorientierten Analyse, Modellierung und Implementierung an Hand eines eigenen Projektes

**Leitfragen:** *Wie realisiere ich meine eigenes Projekt? Welche Klassen und Beziehungen benötige ich?*

### **Vorhabenbezogene Konkretisierung:**

Aufbauend auf den Unterrichtsvorhaben II und III wird die Objektorientierung vertieft und gefestigt durch die Umsetzung eines eigenen einfachen Projektes. Dabei wählen die Schülerinnen und Schüler in kleinen Gruppen selbstständig ein Projektthema aus und erstellen dazu eine Projektplanung. Hierbei liegt der Fokus auf der Erstellung einer funktionierenden Grundversion mit einfachen Methoden und Beziehungen im Rahmen des Erlernten, die noch erweitert (z. B. durch eine grafische Oberfläche oder zusätzliche Methoden) werden kann. Vom Erstellen des Klassen- (mit allen nötigen Attribute und Methoden) und Implementationsdiagramms bis zur Umsetzung und Dokumentation führen die Gruppen die Arbeiten selbst durch. Abschließend stellt jede Gruppe ihr Projekt vor.

**Inhaltliche Schwerpunkte:** Objekte und Klassen, Syntax und Semantik einer Programmiersprache, Analyse, Entwurf und Implementierung einfacher Algorithmen

### **Inhaltsfelder:**

Algorithmen, Daten und ihre Strukturierung, Formale Sprachen und Automaten, Informationssysteme

**Zeitbedarf:** ca. 15 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>Erstellen eines eigene Projektes:</p> <ul style="list-style-type: none"> <li>a) Auswahl des Projektthemas</li> <li>b) Erstellen eines Projektplans</li> <li>c) Erstellen der Klassendiagramme und des Implementationsdiagramms</li> <li>d) Implementation des Projektes</li> <li>e) Dokumentation und Präsentation</li> </ul>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern eine objektorientierte Modellierung (A)</li> <li>• analysieren und erläutern einfache Algorithmen und Programme (A)</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M)</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M)</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M)</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M)</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I)</li> <li>• implementieren einfache Algorithmen</li> </ul>	<p>Mögliche Projekte:</p> <ul style="list-style-type: none"> <li>• Einfache Spiel (ohne Grafik) wie Tic, Tac, Toe oder Mineswepeers</li> <li>• Datenverwaltung wie Jahreskalender, Telefonbuch, Produktverwaltung und Angebotserstellung</li> <li>• Vokabel- oder Rechentrainer, Taschenrechner</li> </ul>

	<p>unter Beachtung der Syntax und Semantik einer Programmiersprache (I)</p> <ul style="list-style-type: none"><li>• interpretieren Fehlermeldungen und korrigieren den Quellcode(I)</li><li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)</li><li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</li><li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)</li><li>• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K)</li></ul>	
--	---	--

## **Unterrichtsvorhaben Nr. E-VI**

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

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

**Vorhabenbezogene Konkretisierung:**

Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Die erste Unterrichtssequenz („Informatiker verändern die Welt – Auswirkungen der Informationstechnologie auf Mensch und Gesellschaft“) können von den Schülerinnen und Schülern als Referate oder in Kleingruppenarbeit präsentiert werden.

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen und auf schülernahe Beispielsituationen zur Anwendung gebracht.

**Inhaltliche Schwerpunkte:** Meilensteine der Informationstechnik, Informatik und Ethik, Automatisierung in der Arbeitswelt, Automatisierung aus Sicht der Informatik, Automatisierung im Alltag, Der gläserne Deutsche, Datenschutz als Grundrecht?, Datenschutz konkret – NSA und Social Media

**Inhaltsfelder:**

Inhaltsfeld (Informatik, Mensch und Gesellschaft)

**Zeitbedarf:** ca. 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p><b>1. Informatiker verändern die Welt – Auswirkungen der Informationstechnologie auf Mensch und Gesellschaft</b></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>Beispiele</p> <ul style="list-style-type: none"> <li>• Meilensteine der Informationstechnik [Referate]</li> <li>• Informatik und Ethik: Textauswertung</li> <li>• Filmbesprechung (Her, Surrogates, War Games)</li> </ul>
<p><b>2. Mensch und Technik – Wer hat die Oberhand?</b></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren anhand von Fallbeispielen die Auswirkungen von Rationalisierung, Digitalisierung und Automatisierung auf den Einzelnen und die Gesellschaft (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li> </ul>	<p>Beispiele:</p> <ul style="list-style-type: none"> <li>• Automatisierung in der Arbeitswelt: Textauswertung, Internetrecherche</li> <li>• Automatisierung aus Sicht der Informatik: Textauswertung, Internetrecherche</li> <li>• Automatisierung im Alltag: Textauswertung, Internetrecherche</li> <li>• Filmbesprechung (2001 – Odyssee im Weltraum, Terminator, Blade Runner, Matrix)</li> </ul>
<p><b>3. Datenschutz – Der gläserne Mensch</b></p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• erkennen anhand von Fallbeispielen die Bedeutung von Datenschutz und Datensicherheit und beurteilen daraus resultierende Chancen und Risiken (A),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li> </ul>	<p>Beispiele:</p> <ul style="list-style-type: none"> <li>• Der gläserne Deutsche (Projekteinstieg)</li> <li>• Datenschutz als Grundrecht?</li> <li>• Datenschutz konkret – NSA und Social Media</li> <li>• Filmbesprechung (1984, Minority Report, Das Leben der Anderen)</li> </ul>