

---

**Schulinterner Lehrplan  
zum Kernlehrplan für die gymnasiale Oberstufe  
am Hüffertgymnasium Warburg**

**Informatik**

---

## Übersichtsraster Unterrichtsvorhaben

### I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p><b>Thema:</b> <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> 5 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p><b>Thema:</b> <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"><li>• Modellieren</li><li>• Implementieren</li><li>• Darstellen und Interpretieren</li><li>• Kommunizieren und Kooperieren</li></ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"><li>• Daten und ihre Strukturierung</li><li>• Formale Sprachen und Automaten</li></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> 6 Stunden</p>

## Einführungsphase

### Unterrichtsvorhaben E-III

**Thema:**

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

**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- 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:** 14 Stunden

### Unterrichtsvorhaben E-IV

**Thema:**

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

**Zentrale Kompetenzen:**

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

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

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

**Zeitbedarf:** 14 Stunden

<b>Einführungsphase</b>	
<p><u>Unterrichtsvorhaben E-V</u></p> <p><b>Thema:</b> <i>Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</i></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> </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> </ul> <p><b>Zeitbedarf:</b> 7 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></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> 12 Stunden</p>
<b>Summe Einführungsphase: 58</b>	

---

## I) Einführungsphase

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

Die Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),
- präsentieren Arbeitsabläufe und -ergebnisse (K),
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).

### 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 müssen.

Zunächst wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Anschließend wird auf die Übertragung von 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:** 5 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Information, deren Kodierung und Speicherung</b></p> <p>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</p> <p>(b) Darstellung von Informationen in Schrift, Bild und Ton</p> <p>(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</p> <p>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),</li> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).</li> </ul>	<p><i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Wingdings, ASCII)</p> <p><i>Beispiel:</i> Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken</p>
<p><b>2. Informations- und Datenübermittlung in Netzen</b></p> <p>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</p> <p>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des</p>		<p><i>Beispiel:</i> Entwicklung von Datenübertragungsstrategien mithilfe von bitweiser Kommunikation über kleine Programme.</p> <p><i>Beispiel:</i> Aufbau von Rechnernetzen mithilfe einer Simulationssoftware (Filius)</p>

<p>Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</p> <p>(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</p> <p>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>		
<p><b>3. Aufbau informatischer Systeme</b></p> <p>(a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“</p> <p>(b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“</p>		<p><i>Material:</i> Demonstrationshardware</p> <p>Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an.</p>

---

## Unterrichtsvorhaben EF-II

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

**Leitfrage:** *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation 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 Objektkarten, Klassenkarten oder Beziehungsdiagramme eingeführt.

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

**Zeitbedarf:** 6 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
<b>1. Identifikation von Objekten</b> (a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt. (b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden ver-	Die Schülerinnen und Schüler <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,</li></ul>	<i>Beispiel:</i> Vogelschwarm Schülerinnen und Schüler betrachten einen Vogelschwarm als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können.  <i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator -



<p>sehen.</p> <p>(c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters</p>	<p>ihren Methoden und Assoziationsbeziehungen (M),</p> <ul style="list-style-type: none"> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• stellen den Zustand eines Objekts dar (D).</li> </ul>	<p>Allgemeine Objektorientierung (<a href="#">Download EF-II.1</a>)</p>
<p><b>2. Analyse von Klassen didaktischer Lernumgebungen</b></p> <p>(a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</p> <p>(b) Teilanalyse von vorgegebenen Klassen für einen grafischen Zugang zur Programmierung.</p>		<p><i>Materialien:</i> Einige vorgefertigte Klassen mit Kreis, Quadrat, Dreieck.</p>
<p><b>3. Implementierung zweidimensionaler, statischer Szenen</b></p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Deklaration und Initialisierung von Objekten</p> <p>(c) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung)</p>		<p><i>Beispiel:</i> Skulpturengarten Schülerinnen und Schüler erstellen ein Programm, das mit Hilfe von geometrischen Objekten einen Skulpturengarten auf den Bildschirm bringt.</p> <p><i>Beispiel:</i> Olympische Ringe Die Schülerinnen und Schüler bilden das Emblem der olympischen Spiele nach.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Sequenzielle Programmierung (<a href="#">Download EF-II.3</a>)</p>

---

## Unterrichtsvorhaben EF-III

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

**Leitfragen:** *Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche unter Berücksichtigung von Tastatureingaben realisieren?*

### **Vorhabenbezogene Konkretisierung:**

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben des Benutzers gesteuerte Animationen aufweisen. Zunächst wird ein Projekt bearbeitet, das in Anlehnung an das vorangegangene Unterrichtsvorhaben eine Szene darstellt, die lediglich aus Objekten besteht, zu denen Klassen vorgegeben sind. Einzelne Objekte der Szene werden animiert, um ein einfaches Spiel zu realisieren oder die Szene optisch aufzuwerten. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt und eingeführt.

Sind an einem solchen Beispiel im Schwerpunkt Schleifen und Verzweigungen eingeführt worden, sollen diese Konzepte an weiteren Beispielprojekten eingeübt werden. Dabei muss es sich nicht zwangsläufig um solche handeln, bei denen Kontrollstrukturen lediglich zur Animation verwendet werden. Auch die Erzeugung größerer Mengen grafischer Objekte und deren Verwaltung in einem Feld kann ein Anlass zur Verwendung von Kontrollstrukturen sein.

Das Unterrichtsvorhaben schließt mit einem Projekt, das komplexere grafische Elemente beinhaltet, so dass die Schülerinnen und Schüler mehr als nur die Klasse erstellen müssen, welche die Szene als Ganzes darstellt. Elemente der Szene müssen zu sinnhaften eigenen Klassen zusammengefasst werden, die dann ihre eigenen Attribute und Dienste besitzen. Auch dieses Projekt soll eine Animation, ggf. im Sinne einer Simulation, sein, bei der Attributwerte von Objekten eigener Klassen verändert werden und diese Veränderungen optisch sichtbar gemacht werden.

Komplexere Assoziationsbeziehungen zwischen Klassen werden in diesem Unterrichtsvorhaben zunächst nicht behandelt. Sie stellen den Schwerpunkt des folgenden Vorhabens dar.

**Zeitbedarf:** 14 Stunden

## Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p><b>1. Bewegungsanimationen am Beispiel einfacher grafischer Objekte</b></p> <p>(a) Kontinuierliche Verschiebung eines Objekts mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife</p> <p>(c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(d) Berechnung von Abständen zwischen Objekten mit Hilfsvariablen</p> <p>(e) Meldungen zur Kollision zweier Objekte mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>• analysieren und erläutern einfache Algorithmen und Programme (A),</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> </ul>	<p><i>Beispiel:</i> Wurfspiel Die Schülerinnen und Schüler realisieren mit Objekten ein Spiel, bei dem ein Ball über den Bildschirm bewegt und auf eine runde Zielscheibe geworfen werden soll.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Kontrollstrukturen (<a href="#">Download EF-III.1</a>)</p>
<p><b>2. Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte</b></p> <p>(a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife)</p> <p>(b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays)</p> <p>(c) Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden</p> <p>(d) Vertiefung: Verschiedene Feldbeispiele</p>	<ul style="list-style-type: none"> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</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 Metho-</li> </ul>	<p><i>Beispiel:</i> Hubschrauberlandeplatz Die Schülerinnen und Schüler realisieren einen runden Hubschrauberlandeplatz und eine Reihe von Landemarkierungen, die in einem Feld verwaltet werden. Mit Hilfe der Landemarkierungen werden verschiedene Lauflichter realisiert.</p> <p><i>Beispiel:</i> Schachbrett Die Schülerinnen und Schüler realisieren mit Hilfe mehrerer Quader ein Schachbrett.</p> <p><i>Beispiel:</i> Magischer Würfel Die Schülerinnen und Schüler erstellen einen</p>

	<p>denaufrufen (I),</p> <ul style="list-style-type: none"> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).</li> </ul>	<p>großen Würfel, der aus mehreren kleineren, farbigen Würfeln besteht.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Kontrollstrukturen (<a href="#">Download EF-III.2</a>)</p>
<p><b>3. Modellierung und Animation komplexer grafisch repräsentierbarer Objekte</b></p> <p>(a) Modellierung eines Simulationsprogramms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen Objekten zeigen mit Hilfe eines Implementationsdiagramms</p> <p>(b) Implementierung eigener Methoden mit und ohne Parameterübergabe</p> <p>(c) Realisierung von Zustandsvariablen</p> <p>(d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten</p> <p>(e) Animation mit Hilfe des Aufrufs von selbstimplementierten Methoden</p> <p>(f) Vertiefung: Weitere Projekte</p>		<p><i>Beispiel: Kerzensimulation</i> Die Schülerinnen und Schüler modellieren und erstellen eine Klasse, mit deren Hilfe Kerzen simuliert werden können. Eine Kerze kann angezündet und gelöscht werden. Abgesehen davon brennen Kerzen abhängig von ihrer Dicke unterschiedlich schnell ab.</p> <p><i>Beispiel: Billardtisch</i> Die Schülerinnen und Schüler erstellen eine Simulation mehrerer Kugeln, die über einen Billardtisch rollen.</p> <p><i>Beispiel: Ampeln</i> Die Schülerinnen und Schüler erstellen eine Ampelkreuzung mit mehreren Ampelanlagen an einem Bahnübergang.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Eigene Klassen (<a href="#">Download EF-III.3</a>)</p>

---

## Unterrichtsvorhaben EF-IV

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

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

### **Vorhabenbezogene Konkretisierung:**

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

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

Zum Abschluss kann kurz auf das Prinzip der abstrakten Klasse eingegangen werden. Dieser Inhalt ist aber nicht obligatorisch für die Einführungsphase.

**Zeitbedarf:** 14 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
<b>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</b> (a) Einführung der Objektselektion mit der Maus	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>• analysieren und erläutern eine objektorientierte Modellierung (A),</li></ul>	<i>Beispiel:</i> Seifenblasen Die Schülerinnen und Schüler entwickeln ein Spiel, bei dem Seifenblasen über den Bildschirm schweben und durch anklicken mit der Maus zum Zerplatzen gebracht werden können.

<p>(b) Einführung der Klasse GrafikObjekt als Oberklasse aller sichtbaren Objekte</p> <p>(c) Steuerung einfacher grafischer Objekte über eine Referenz aktuell, die jeweils durch eine Klickselektion mit der Maus auf ein neues Objekt gesetzt werden kann.</p>	<ul style="list-style-type: none"> <li>• stellen die Kommunikation zwischen Objekten grafisch dar (M),</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen durch Beschrei-</li> </ul>	<p><i>Beispiel:</i> Sonnensystem Die Schülerinnen und Schüler entwickeln eine Simulation des Sonnensystems bei der Daten zum angeklickten Planeten ausgegeben werden.</p> <p><i>Materialien:</i> -</p>
<p><b>2. Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten</b></p> <p>(a) Modellierung des Spiels ohne Berücksichtigung der Kollision mit Hilfe eines Implementationsdiagramms</p> <p>(b) Dokumentation der Klassen des Projekts</p> <p>(c) Implementierung eines Prototypen ohne Kollision</p> <p>(d) Ergänzung einer Kollisionsabfrage durch zusätzliche Assoziationsbeziehungen in Diagramm, Dokumentation und Quellcode</p> <p>(e) Verallgemeinerung der neuen Verwendung von Objektreferenzen</p> <p>(f) Vertiefung: Entwicklung weiterer Spiele und Simulationen mit vergleichbarer Grundmodellierung</p>	<ul style="list-style-type: none"> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>• modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>• testen Programme schrittweise anhand von Beispielen (I),</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>• modifizieren einfache Algorithmen und Programme (I),</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>• dokumentieren Klassen durch Beschrei-</li> </ul>	<p><i>Beispiel:</i> Ufospiel Die Schülerinnen und Schüler entwickeln die Simulation eines Ufos, das Asteroiden ausweichen soll mit denen eine Kollision möglich ist.</p> <p><i>Beispiel:</i> Billardkugeln Die Schülerinnen und Schüler entwickeln ein Spiel, bei dem tickende Billardkugeln mit einer beweglichen Box eingefangen werden sollen.</p> <p><i>Beispiel:</i> Autospiel Die Schülerinnen und Schüler entwickeln ein Autospiel, bei dem ein Auto durch einen Wald fahren und mit Bäumen kollidieren kann.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Assoziationen (<a href="#">Download EF-IV.1</a>)</p> <p>Informationsblatt: Implementationsdiagramme (<a href="#">Download EF-IV.2</a>)</p>
<p><b>3. Erarbeitung einer Simulation mit grafischen Objekten, die sich durch un-</b></p>	<ul style="list-style-type: none"> <li>• dokumentieren Klassen durch Beschrei-</li> </ul>	<p><i>Beispiel:</i> Schneemann Die Schülerinnen und Schüler erstellen eine</p>

<p><b>terschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden)</b></p> <ul style="list-style-type: none"> <li>(a) Analyse und Erläuterung einer Basisversion der grafischen Klasse</li> <li>(b) Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode)</li> <li>(c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</li> </ul>	<p>bung der Funktionalität der Methoden (D).</p>	<p>Simulation von Schneemännern, die unterschiedliche Kopfbedeckungen tragen.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Vererbung (<a href="#">Download EF-IV.3</a>)</p>
<p><b>4. Entwicklung einer komplexeren Simulation mit grafischen Elementen, die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden)</b></p> <ul style="list-style-type: none"> <li>(a) Analyse und Erläuterung einer einfachen grafischen Animationsklasse</li> <li>(b) Spezialisierung der Klasse zu Unterklassen mit verschiedenen Animationen durch Überschreiben der entsprechenden Animationsmethode</li> <li>(c) Reflexion des Prinzips der späten Bindung</li> <li>(d) Vertiefung: Entwicklung eines vergleichbaren Projekts mit einer (abstrakten) Oberklasse</li> </ul>		<p><i>Beispiel:</i> Flummibälle Die Schülerinnen und Schüler entwickeln eine Simulation von Flummibällen, bei der unterschiedliche Bälle unterschiedliche Bewegungen durchführen.</p> <p><i>Beispiel:</i> Weihnachtsbaum Die Schülerinnen und Schüler entwickeln eine Simulation eines Weihnachtsbaums mit Hilfe einer abstrakten Klasse Schmuck.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Vererbung (<a href="#">Download EF-IV.4</a>)</p>

---

## Unterrichtsvorhaben EF-V

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

**Leitfragen:** *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

### 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 auch verzichtet werden kann.

Zunächst erarbeiten die Schülerinnen und Schüler mögliche Einsatzszenarien für Such- und Sortieralgorithmen, um sich der Bedeutung einer effizienten Lösung dieser Probleme bewusst zu werden. Anschließend werden Strategien zur Sortierung mit Hilfe eines explorativen Spiels von den Schülerinnen und Schülern selbst erarbeitet und hinsichtlich der Anzahl notwendiger Vergleiche auf ihre Effizienz untersucht.

Daran anschließend werden die erarbeiteten Strategien systematisiert und im Pseudocode notiert. Die Schülerinnen und Schüler sollen auf diese Weise das *Sortieren durch Vertauschen*, das *Sortieren durch Auswählen* und mindestens einen weiteren Sortieralgorithmus, kennen lernen.

Des Weiteren soll das Prinzip der *binären Suche* behandelt und nach Effizienz Gesichtspunkten untersucht werden.

**Zeitbedarf:** 7 Stunden

### Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<b>1. Explorative Erarbeitung eines Sortierverfahrens</b> (a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B.	Die Schülerinnen und Schüler <ul style="list-style-type: none"><li>beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren</li></ul>	<i>Beispiel:</i> Sortieren mit simuliertem Speicher und Zahlenkarten Die Schüler bekommen die Aufgabe, Karten mit unterschiedlichen Zahlen auf einem simulierten Speicher zu sortieren.



<p>Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p>	<p>hinsichtlich Zeit und Speicherplatzbedarf (A),</p> <ul style="list-style-type: none"> <li>entwerfen einen weiteren Algorithmus zum Sortieren (M),</li> <li>analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).</li> </ul>	<p><i>Materialien:</i></p> <p>Computer science unplugged – Sorting Algorithms, URL: <a href="http://www.csunplugged.org/sorting-algorithms">www.csunplugged.org/sorting-algorithms</a> abgerufen: 30. 03. 2014</p>
<p><b>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>(f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)</p>		<p><i>Beispiele:</i> Sortieren durch Auswählen, Sortieren durch Vertauschen, Quicksort</p> <p>Quicksort ist als Beispiel für einen Algorithmus nach dem Prinzip <i>Teile und Herrsche</i> gut zu behandeln. Kenntnisse in rekursiver Programmierung sind nicht erforderlich, da eine Implementierung nicht angestrebt wird.</p> <p><i>Materialien:</i></p> <p>Computer science unplugged – Sorting Algorithms, URL: <a href="http://www.csunplugged.org/sorting-algorithms">www.csunplugged.org/sorting-algorithms</a> abgerufen: 30. 03. 2014</p>
<p><b>3. Binäre Suche auf sortierten Daten</b></p> <p>(a) Suchaufgaben im Alltag und im Kon-</p>		<p><i>Beispiel:</i> Simulationsspiel zur binären Suche nach Tischtennisbällen</p>

---

<p>text informatischer Systeme</p> <ul style="list-style-type: none"><li>(b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche</li><li>(c) Effizienzbetrachtungen zur binären Suche</li></ul>		<p>Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p> <p><i>Materialien:</i> Computer science unplugged – Searching Algorithms, URL: <a href="http://www.csunplugged.org/searching-algorithms">www.csunplugged.org/searching-algorithms</a>, abgerufen: 30. 03. 2014</p>
---	--	--

---

## Unterrichtsvorhaben EF-VI

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

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

### **Vorhabenbezogene Konkretisierung:**

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

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

**Zeitbedarf:** 12 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Beispiele, Medien, Materialien</b>
<b>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</b> (a) Mögliche Themen zur Erarbeitung in Kleingruppen: <ul style="list-style-type: none"><li>„Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“</li></ul>	Die Schülerinnen und Schüler <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</li></ul>	<i>Beispiel:</i> Ausstellung zu informatischen Themen Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können.

<ul style="list-style-type: none"> <li>• „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“</li> <li>• „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“</li> <li>• „Kodieren von Texten und Bildern: ASCII, RGB und mehr“</li> <li>• „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“</li> </ul> <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<p>tung (A),</p> <ul style="list-style-type: none"> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen (D),</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).</li> </ul>	<p><i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>
<p><b>2. Vertiefung des Themas Datenschutz</b></p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz (<a href="#">Download EF-VI.1</a>)</p>

---

## Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

### Überfachliche Grundsätze:

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

### Fachliche Grundsätze:

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

---

## Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Hüffertgymnasiums 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.

### Beurteilungsbereich Klausuren

#### Verbindliche Absprachen:

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

#### Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr  
Dauer der Klausur: 90 Minuten
- Grundkurse Q 1: 2 Klausuren je Halbjahr  
Dauer der Klausuren: 90 Minuten
- Grundkurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 135 Minuten
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1 eine Facharbeit geschrieben werden.

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

---

## Kriterien

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

Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsumme zu den Notenstufen an dem Zuordnungsschema des Zentralabiturs.

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

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

## Beurteilungsbereich Sonstige Mitarbeit

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

## Leistungsaspekte

### Mündliche Leistungen

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

### Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen

### Sonstige schriftliche Leistungen

- Lernerfolgsüberprüfung durch kurze schriftliche Übungen (optional)
- Bearbeitung von schriftlichen Aufgaben im Unterricht

---

## Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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



---

## Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

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

---

## Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

### **Zusammenarbeit mit anderen Fächern**

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden.

### **Vorbereitung auf die Erstellung der Facharbeit**

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung.

### **Qualitätssicherung und Evaluation**

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

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

Das schulinterne Curriculum ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmalig nach Ende der Einführungsphase im Sommer 2015, 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 Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.