



Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe

Informatik

(Stand: 24.09.2014)

Dieser schulinterne Lehrplan ist vorerst für die Einführungsphase formuliert und wird in den kommenden Schuljahren um Angaben für die Qualifikationsphase (Q1 und Q2) ergänzt werden

Inhalt

	Seite
1 Entscheidungen zum Unterricht	3
1.1 Unterrichtsvorhaben	3
1.1.1 Übersichtsraster der Unterrichtsvorhaben	5
1.1.2 Konkretisierte Unterrichtsvorhaben	8
1.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	25
1.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	26
2 Qualitätssicherung und Evaluation	29

1 Entscheidungen zum Unterricht

Hinweis: Die nachfolgend dargestellte Umsetzung der verbindlichen Kompetenzerwartungen des Kernlehrplans findet auf zwei Ebenen statt. Das **Übersichtsraster** gibt den Lehrkräften einen raschen Überblick über die laut Fachkonferenz verbindlichen Unterrichtsvorhaben pro Schuljahr. In dem Raster sind außer dem Thema des jeweiligen Vorhabens das schwerpunktmäßig damit verknüpfte Inhaltsfeld bzw. die Inhaltsfelder, inhaltliche Schwerpunkte des Vorhabens sowie Schwerpunktkompetenzbereiche ausgewiesen. Die **Konkretisierung von Unterrichtsvorhaben** führt weitere Kompetenzerwartungen auf und verdeutlicht vorhabenbezogene Absprachen, z.B. zur Festlegung auf einen Aufgabentyp bei der Lernerfolgsüberprüfung durch eine Klausur.

1.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 1.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 1.1.2) 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 1.2 bis 1.3 zu entnehmen sind.

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

1.1.1 Übersichtsraster der Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema: <i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Informatiksysteme• Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Einzelrechner• Dateisystem• Internet• Einsatz von Informatiksystemen <p>Zeitbedarf: 6 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 8 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: 18 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: 24 Stunden

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema: <i>Projektartiges Arbeiten anhand kontextbezogener Beispiele</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementierung einfacher Klassen und Klassenbeziehungen • Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema: <i>Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatik, Mensch und Gesellschaft • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Wirkungen der Automatisierung • Geschichte der automatischen Datenverarbeitung • Schnittstelle Mensch-Computer <p>Zeitbedarf: 12 Stunden</p>
Summe Einführungsphase: 80	

2.1.2 Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die im *Unterkapitel 2.1.1* aufgeführten Unterrichtsvorhaben konkretisiert werden.

Hinweis:

Verbindliche Festlegungen der Fachkonferenz:

Die Fachkonferenz des St.-Ursula-Gymnasiums Attendorn hat Themen, Leitfragen und die Ausführungen unter der Überschrift *Vorhabenbezogene Konkretisierung* verbindlich vereinbart, ebenso die Sequenzierung der Unterrichtsvorhaben (erste Tabellenspalte) und die ausgewiesenen Kompetenzen (zweite Tabellenspalte). Alle Mitglieder der Fachkonferenz haben sich darauf verständigt, in ihrem Unterricht Lerngelegenheiten anzubieten, so dass Schülerinnen und Schüler diese Kompetenzen im Rahmen der festgelegten Unterrichtssequenzen erwerben oder vertiefen können.

Unterrichtliche Anregungen:

Die angeführten Beispiele, Medien und Materialien sind dagegen Vorschläge bzw. Hilfen für die Lehrkräfte unserer Schule. In diesen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich.

In der Einführungsphase wird die didaktische Bibliothek SuM („Stifte und Mäuse“) verwendet. Die folgenden Installationspakete und Dokumentationen stehen zur Verfügung:

- Installationspaket (Windows) [\(Download\)](#)

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse

<http://www.standardsicherung.schulministerium.nrw.de/abitur-gost/fach.php?fach=15> (abgerufen: 30. 04. 2014)

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.

Abschließend wird die Installation der konkreten Arbeitsumgebung BlueJ mit JAVA SE und der SuM-Bibliothek besprochen. Somit soll sichergestellt werden, dass alle Schüler/innen in der gewohnten Arbeitsumgebung auch zu Hause (weiter-)arbeiten können.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Information, deren Kodierung und Speicherung</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"> • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). 	<p><i>Beispiel:</i> Netzwerkarchitektur in der Schule</p>

<p>2. Informations- und Datenübermittlung in Netzen</p> <ul style="list-style-type: none"> (a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation (b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server) (c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll) (d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet 		<p><i>Beispiel:</i> Rollenspiel zur Paketvermittlung im Internet</p> <p>Schülerinnen und Schüler übernehmen die Rollen von Clients und Routern. Sie schicken spielerisch Informationen auf Karten von einem Schüler-Client zum anderen. Jede Schülerin und jeder Schüler hat eine Adresse, jeder Router darüber hinaus eine Routingtabelle. Mit Hilfe der Tabelle und einem Würfel wird entschieden, wie ein Paket weiter vermittelt wird.</p>
<p>3. Aufbau informatischer Systeme</p> <ul style="list-style-type: none"> (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“ 		<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 Beziehungsdigramme eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe der didaktischen Programmierumgebung SuM („Stifte und Mäuse“) 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.

Da bei der Umsetzung dieser ersten Projekte konsequent 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 führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Identifikation von Objekten</p> <p>(a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen.</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>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), stellen die Kommunikation zwischen Objekten grafisch dar (M), implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), stellen den Zustand eines Objekts dar (D). 	<p><i>Beispiel:</i> Computerfigur Super-Mario</p> <p>Schülerinnen und Schüler betrachten eine Spielfigur als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können. Super-Mario wird mit seinen Eigenschaften und mit seinem Können als Instanz der Klasse analysiert.</p>
<p>2. Analyse von Klassen didaktischer Lernumgebungen</p> <p>(a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen)</p> <p>(b) Teilanalyse der Klassen der didaktischen Lernumgebungen SuM</p>		<p><i>Materialien:</i> Dokumentation der didaktischen Bibliothek SuM</p>
<p>3. Implementierung zweidimensionaler, statischer Szenen</p> <p>(a) Grundaufbau einer Java-Klasse</p> <p>(b) Konzeption einer Bildschirmland-</p>		<p><i>Beispiel:</i> Skulpturengarten</p> <p>Schülerinnen und Schüler erstellen ein Programm, das mit Hilfe von geometrischen Objekten eine Bildschirmlandschaft erzeugt</p>

<p>schaft mit verschiedenen, sichtbaren Objekten</p> <p>(c) Deklaration und Initialisierung von Objekten</p> <p>(d) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung)</p>		<p><i>Beispiel:</i> Programmgesteuertes Zeichnen Die Schülerinnen und Schüler erstellen ein Programm, das mit Hilfe der grundlegenden Objekte Bildschirm und Stift geometrische Figuren auf dem Bildschirm darstellt.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Sequenzielle Programmierung</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 grafischer Objekte, auch unter Berücksichtigung von Tastatureingaben realisieren?*

Vorhabenbezogene Konkretisierung:

Der Schwerpunkt dieses Unterrichtsvorhabens liegt auf der Entwicklung mehrerer Projekte, die durch Eingaben oder Aktionen 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 das didaktische System Klassen vorgibt. 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

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: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Bewegungsanimationen am Beispiel eines Dartpfeiles</p> <p>(a) Kontinuierliche des Dartpfeiles mit Hilfe einer Schleife (While-Schleife)</p> <p>(b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife, z.B. Drehen des Pfeiles</p> <p>(c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen</p> <p>(d) Berechnung von Abständen zwischen gezeichneten Objekten mit Hilfsvariablen</p> <p>(e) Meldungen zur Kollision des Dartpfeiles und der Scheibe mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich 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), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), 	<p><i>Beispiel:</i> Pfeilwurf Die Schülerinnen und Schüler realisieren mit Objekten der SuM-Umgebung ein Spiel, bei dem ein Dartpfeil über den Bildschirm bewegt und auf eine runde Zielscheibe geschossen werden soll.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Kontrollstrukturen</p>
<p>2. Erstellen und Verwalten von Objekten, Modellierung und Animation von Objekten</p> <p>(a) Steuerung von Objekten mittels kopfgesteuerten und fußgesteuerten Schleifen</p> <p>(b) Spezialisierung von Objekten mittels IST-Beziehungen und Unterklassenstruktur</p> <p>(c) Steuerung von Objekten mittels Tas-</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich 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), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), 	<p><i>Beispiel:</i> Freihandzeichnen Die Schülerinnen und Schüler realisieren ein Projekt, in welchem mit Objekten der Bibliothek SuM auf dem Bildschirm ein einfaches Zeichenprogramm erstellt wird und bei welchem die Kontrollstrukturen und Verzweigungen vertiefend behandelt werden</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator -</p>

<p>taturabfragen (d) Steuerung von Objekten mittels mehrfacher Verzweigung</p>	<ul style="list-style-type: none"> • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Kontrollstrukturen</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: 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Vertiefung des Referenzbegriffs und Einführung des Prinzips der dynamischen Referenzierung</p> <p>(a) Einführung der Klasse Kreis und Quadrat</p> <p>(b) Einführung der Klasse <code>Figur</code> als Oberklasse aller sichtbaren Figuren</p> <p>(c) Steuerung einfacher grafischer Objekte über eine Referenz <code>aktuelleFigur</code>, die jeweils durch eine Klickselektion mit der Maus bewegt werden kann</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> analysieren und erläutern eine objektorientierte Modellierung (A), stellen die Kommunikation zwischen Objekten grafisch dar (M), ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, 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) 	<p><i>Materialien:</i></p> <p>-</p>
<p>2. Entwicklung eines Spiels mit der Notwendigkeit von Kollisionskontrollen zwischen zwei oder mehr grafischen Objekten</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 Spie-</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, 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) 	<p><i>Beispiel:</i> Billardkugeln Die Schülerinnen und Schüler entwickeln ein Spiel, bei dem tickende Billardkugeln über einen Tisch laufen.</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</p> <p>Informationsblatt: Implementationsdiagramme</p>

<p>le und Simulationen mit vergleichbarer Grundmodellierung</p>	<ul style="list-style-type: none"> • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • modifizieren einfache Algorithmen und Programme (I), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p><i>Beispiel:</i> Schneemann Die Schülerinnen und Schüler erstellen eine Simulation von Schneemännern, die unterschiedliche Kopfbedeckungen tragen.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Vererbung</p>
<p>3. Erarbeitung einer Simulation mit grafischen Objekten, die sich durch unterschiedliche Ergänzungen voneinander unterscheiden (Vererbung durch Spezialisierung ohne Überschreiben von Methoden)</p> <p>(a) Analyse und Erläuterung einer Basisversion der grafischen Klasse</p> <p>(b) Realisierung von grafischen Erweiterungen zur Basisklasse mit und ohne Vererbung (Implementationsdiagramm und Quellcode)</p> <p>(c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</p>		<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 <code>Schmuck</code>.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator – Vererbung</p>

trakten) Oberklasse		
5. Entwicklung einer komplexeren Simulation mit grafischen Elementen, die unterschiedliche Animationen durchführen (Vererbung mit Überschreiben von Methoden) als Vorbereitung der Verkettung		<i>Beispiel: Zugprojekt</i> Die Schülerinnen und Schüler entwickeln ein Zugprojekt, in welchem Nachrichten von einem Waggon auf den anderen weiter geschickt werden und sich durch Verkettung der gesamte Zug bewegt

Unterrichtsvorhaben EF-V

Thema: Projektartiges Arbeiten anhand kontextbezogener Beispiele

Leitfragen: *Wie können komplexere Aufgaben/Projekte in der Gruppe effizient geplant, bearbeitet und fertiggestellt werden?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit komplexeren, jedoch überschaubaren Projekten zur Erstellung einer Software. Der Schwerpunkt des Vorhabens liegt dabei auf den vollständigen Erstellungsprozess einer Software von der von Anwendern formulierte Auftragsbeschreibung über die objektorientierte Analyse (OOA) und Programmierung (OOP) einschließlich notwendiger Tests und Kontrollen bis hin zum fertigen Endprodukt. Dabei wird in der OOA das Prinzip der abstrakten Klassen aufgegriffen.

Zunächst analysieren die Schülerinnen und Schüler die Aufgabenformulierung hinsichtlich der enthaltenen Klassen und Objekte mit ihren Attributen und Methoden, um dieses anschließend in ein objektorientiertes Klassenbeziehungsmodell zu übertragen und dieses in Arbeits- teilung unter besonderer Beachtung evtl. vorkommender Schnittstellen zu implementieren, zu testen und fertig zu stellen.

Anschließend die Phasen eines Softwareprojektes analysiert und bei der Implementation eines zweiten Projektes die erworbenen Strategie- n und Arbeitsmethoden unter zusätzlicher Beachtung der Arbeitsschritte gefestigt und erweitert.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Die Phasen vom Auftrag bis zum End- produkt bei der Softwareerstellung (a) Formulierung und Erläuterung not- wendiger Phasen beim Erstellen von Software; Entwicklung eines Pha- senmodells	Die Schülerinnen und Schüler <ul style="list-style-type: none">gliedern den Prozess der Erstellung von Software in Arbeitsschritte(A)führen ein Pflichtenheft und reflektieren	<i>1. Unterrichtsprojekt:</i> „Einarmiger Bandit“ Alle Phasen bei der Softwareentwicklung (<i>Pflich- tenhefterstellung, Entwurfsphase, Implementati- onsphase, Testphase</i>) werden berücksichtigt. Nachdem eine Ist-Analyse den aktuellen Stand des zu implementierenden Automaten in der Rea-

<p>(b) Exemplarische Anwendung des Modells auf ein Unterrichtsprojekt</p> <p>(c) Arbeitsteilige Entwicklung unter besonderer Beachtung der Schnittstellen zwischen den Teilprodukten</p> <p>(d) Modultest und Zusammenbau der Module</p> <p>(e) Verwendung einer Ereignisbibliothek</p>	<p>daran fortlaufend den Stand der Implementation (D)</p> <ul style="list-style-type: none"> • entwerfen ein Verfahren zur Erstellung eines Klassenmodells auf der Grundlage einer konkreten Aufgabenbeschreibung (A, K) • entwickeln Klassenmodelle (M) • implementieren die Klassen des Projektes arbeitsteilig (I) • testen die selbst erstellten Module mit dem Testwerkzeug von BlueJ (I) • verwenden beim Zusammenbau der Module die zur SuM-Bibliothek gehörenden Ereignis- und Komponentenklassen (I) • unterscheiden im Programm die Ebenen des Datenmodells, der Darstellung und der Steuerung („Model View Controller“) und programmieren diese getrennt voneinander (M) 	<p>lität erfasst, werden in einer Soll-Analyse Oberfläche und Funktionalität des Modells, also des softwaremäßigen Zielproduktes ohne Betrachtung der Implementation festgelegt. Das Zielprodukt wird formuliert und beteiligter Objekte (incl. Attributen und Methoden) analysiert (ggf. Substantiv-Verb-Methode).</p> <p>Nach der Implementation der beteiligten Klassen werden Regressionstests derselben durchgeführt, bevor in einem letzten Schritt das GUI (<i>graphical User Interface</i>) diese mit Hilfe der Ereignis- und Komponentenbibliothek funktionell verbindet.</p> <p><i>Materialien:</i> Schulbuch „Informatik mit Java“, SuM-Bibliotheken (Ereignis und Komponenten), Testwerkzeug (in BlueJ integriert)</p>
<p>2. Drei Schichten eines Programmes – Model View Controller</p> <p>(a) Aufteilung eines Programms in drei Schichten</p> <p>(b) Oberflächenerstellung (GUI) mit Hilfe des integrierten Programmgenerators</p> <p>(c) Analyse benötigter Daten und Datenmodellierung</p> <p>(d) Programmsteuerung mit den Methoden der mitgelieferten Komponenten des Programmgenerators</p> <p>(e) Erzeugung und Verwendung eines Applet zum Programm</p>		<p><i>2. Unterrichtsprojekt: „Bruchrechner“</i></p> <p>Zum ersten Mal entwickeln die Schülerinnen und Schüler Klassen die keine graphische Repräsentation auf dem Bildschirm besitzen. Und dennoch müssen die Berechnungsergebnisse der Klassen des Datenmodells (Model) je nach Eingaben dargestellt bzw. ausgegeben werden (View). Die Methoden der Komponenten auf der graphischen Benutzeroberfläche kontrollieren die Ein- und Ausgabe. Sie sind der Controller und stellen die Schnittstelle zwischen dem Benutzer und den berechnenden Klassen dar. Bei der Umsetzung des Projektes „Bruchrechner“ wird besonderer Wert auf die strikte Trennung dieser Schichten gelegt.</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: 15 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none">„Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“„Eine kleine Geschichte der Kryptogra-	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),	<p><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.</p> <p><i>Beispiel:</i> Die Schülerinnen und Schüler vertiefen die Geschichte der von-Neumann-Architektur, erweitern sie um den Aspekt der Kommunikation</p>

<p>phie: von Caesar zur Enigma“</p> <ul style="list-style-type: none"> • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ • Schnittstelle Mensch-Computer <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<ul style="list-style-type: none"> • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K). 	<p>mit (Hilfe) der EDV-Hardware (mit/ohne Maus, mit Fingern)</p> <p><i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>
<p>2. Vertiefung des Themas Datenschutz</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</p>

1.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des St.-Ursula-Gymnasiums 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 Bedeutsamkeit.
- 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.

1.3 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 St.-Ursula-Gymnasiums 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.

1.3.1 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: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: ggf. 1 Klausur unter Abiturbedingungen
- 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.

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.

1.3.2 Beurteilungsbereich Sonstige Mitarbeit

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

Verbindliche Absprachen der Fachkonferenz

- 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.

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

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.
Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

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

Die Bewertungskriterien stützen sich auf

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

Besonderes Augenmerk ist dabei auf

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

Bei Gruppenarbeiten auch auf

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

Bei Projektarbeit darüber hinaus auf

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

Grundsätze der Leistungsrückmeldung und Beratung

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

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

Die Leistungsrückmeldung kann

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

erfolgen.

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

2 Qualitätssicherung und Evaluation

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

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

Das schulinterne Curriculum (siehe 1.1) 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.