

**Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe
am Wilhelm-Hittorf-Gymnasium, Münster**

Informatik - Einführungsphase

(Entwurfssfassung 04.03.2014 16:23)

1 Vorstellung des Informatikunterrichts am Wilhelm-Hittorf-Gymnasium Münster	3
1.1 Vorstellung der Schule und der Fachgruppe	3
1.2 Medientechnische Grundbildung in der Einführungsphase.....	3
1.3 Informatikunterricht im Wahlpflichtbereich II (Klassen 8+9).....	4
1.4 Informatikunterrichts in der Gymnasialen Oberstufe	5
2 Entscheidungen zum Unterricht	6
2.1 Unterrichtsvorhaben	6
Unterrichtsvorhaben EF-I: Einführung in die Informatik.....	6
Unterrichtsvorhaben EF-II: Einführung in die Informatik – Informatisches Problemlösen und Algorithmen als Beschreibung von Abläufen	8
Unterrichtsvorhaben EF-III: Einführung in die OOP mit Greenfoot	10
Unterrichtsvorhaben EF-IV: Suchen und Sortieren	13
Unterrichtsvorhaben EF-V: Softwareentwicklung.....	15
2.2 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	18
2.2.1 Vorbemerkungen.....	18
2.2.2 Bewertungsbereich Sonstige Mitarbeit.....	19
2.2.3 Klausuren	20
3 Weitere Hinweise und Entscheidungen	21
4 Qualitätssicherung und Evaluation	22
4.1. Kollegialer Austausch.....	22
4.2. Evaluation durch die Schülerinnen und Schüler.....	22
5 Kompetenzerwartung und inhaltliche Schwerpunkte in der Einführungsphase	23
Inhaltsfeld 1: Daten und ihr Strukturierung	23
Inhaltsfeld 2: Algorithmen	25
Inhaltsfeld 3: Formale Sprachen und Automaten	28
Inhaltsfeld 4: Informatiksysteme.....	29
Inhaltsfeld 5: Informatik, Mensch und Gesellschaft.....	31

1 Vorstellung des Informatikunterrichts am Wilhelm-Hittorf-Gymnasium Münster

1.1 Vorstellung der Schule und der Fachgruppe

Beim Wilhelm-Hittorf-Gymnasium handelt es sich um ein vierzütiges Gymnasium mit einer langen Tradition im Bereich der MINT-Fächer, an dem seit mehr als 25 Jahren Informatik unterrichtet wird. Die Fachschaft Informatik besteht zurzeit aus vier Kollegen, die alle ein grundständiges Informatikstudium und Referendariat durchlaufen haben.

In drei voll ausgestatteten Rechnerräumen mit jeweils 16 Arbeitsplätzen kann der Unterricht durchgeführt werden. Ein Selbstlernzentrum ermöglicht es den Schülerinnen und Schülern in der GOST, auch außerhalb der Unterrichtszeiten in der gewohnten Umgebung zu arbeiten.

Informatische Inhalte und Fragestellungen werden am Wilhelm-Hittorf-Gymnasium bereits in den Jahrgangsstufen 5 und 7 in der **MEDIEN-TECHNISCHEN GRUNDBILDUNG**, im Wahlpflichtbereich II der Jahrgangsstufen 8 und 9 im **FACH INFORMATIK** (mit 3 Schulstunden pro Halbjahr) und in der Gymnasialen Oberstufe in **GRUND- UND LEISTUNGSKURSEN** behandelt. [siehe Grafik rechts].

Jgst 5	Medientechnische Grundbildung (ganzjährig mit einer Wochenstunde)	
Jgst 6		
Jgst 7	Medientechnische Grundbildung (halbjährig mit einer Wochenstunde)	
Jgst 8	Informatik WP II (ganzjährig mit 2,5 Wochenstunden)	andere Fächer im WP II
Jgst 9	Informatik WP II (ganzjährig mit 2 Wochenstunde)	
EF	Informatik „fortg.“	Informatik „neu“
Q1	Grundkurs oder Leistungskurs	
Q2	Grundkurs oder Leistungskurs	
Abitur	mdl. oder schriftl. im Grundkurs	schriftliches Abitur im Leistungskurs

1.2 Medientechnische Grundbildung in der Einführungsphase

Klassenbezogen und damit für alle Schülerinnen und Schüler verpflichtend werden in den Klassen 5 und 7 Themen der technisierten Kommunikation, der automatisierten Informationsbeschaffung und –verarbeitung und der Chancen und Gefahren in der vernetzten Welt thematisiert. „Ziel ist es, eine zeitgemäße und fachlich substantielle informatische Bildung in den Schulen zu fördern“¹, die über eine reine Anwendungsschulung von Programmen hinausgeht.

¹ „Grundsätze und Standards für Informatik in der Schule“ – Bildungsstandards für die Sekundarstufe I - Empfehlung der Gesellschaft für Informatik e.V. vom 24. Januar 2008 (online erhältlich unter <http://www.informatikstandards.de>)

Vielmehr sollen im Unterricht Inhalte gelernt und Prozesskompetenzen erworben werden, die zu einem produktiven, reflektierten und kritischen Umgang mit Medien führen und die erworbenen Kenntnisse unabhängig eines Produktes oder einer Produktgruppe anwendbar machen.

Damit folgt die Konzeption für die medientechnische Grundbildung am Wilhelm-Hittorf-Gymnasium aktuellen fachdidaktischen Diskussionen und orientiert sich

- an den „Grundsätzen und Standards für die Informatik in der Sek 1“²
- am Bewusstsein eines „Dienstleistungscharakters“ für andere Fächer im Fächerkanon des WHG und
- an der Struktur des etablierten Informatikunterrichts ab der Jahrgangsstufe 8 am WHG.

Eine Besonderheit ist dabei sicherlich, dass in der Einführungsphase der Gymnasialen Oberstufe ähnlich wie aus den Fremdsprachen bekannt, Neueinsteiger in getrennten Kursen zu den Schülerinnen und Schüler, die bereits im Wahlpflichtbereich II Informatikunterricht hatten, unterrichtet werden.

Zu beachten ist, dass am Wilhelm-Hittorf-Gymnasium der Unterricht im 60-Minuten-Modell durchgeführt wird, wobei im Leistungskurs einmal in der Woche eine doppelte Zeitstunde unterrichtet wird. Dies bedeutet für den Informatikunterricht, dass sich in jeder Stunde theoretische und praktische Phasen abwechseln und im Sinne eines problemorientierten Unterrichts der Problemaufwurf, die Problemlösung und die Präsentation oftmals in einer Stunde erledigt werden können.

1.3 Informatikunterricht im Wahlpflichtbereich II (Klassen 8+9)

Aufbauend auf den Kompetenzen, die die Schülerinnen und Schüler in der medientechnischen Grundbildung erlernt haben, werden im Wahlpflichtbereich der Klassen 8 und 9 informatische Fragestellungen vertiefend behandelt. Der Inhaltsbereich Algorithmen gewinnt hier mehr an Bedeutung.

Anhand der dargestellten Themenbereiche bilden die Schülerinnen und Schüler Kompetenzen in der Informatik aus, wobei die Themen so gewählt sind, dass ein Neueinstieg in das Fach Informatik in der Oberstufe möglich ist. Die behandelten Themenbereiche, die aufgrund schulorganisatorischer Abläufe in unterschiedlichen Reihenfolgen bearbeitet werden können, sehen in den vier Halbjahren wie im schulinternen Lehrplan³ veröffentlicht aus (Robotik – Technisierung unserer Welt, Kryptologie – Verschlüsseln von Caesar bis heute, Das Internet – Mediennutzung, Mediengestaltung und Medienkritik, logische Schaltungen, Datenbanken).

² ebd.

³ Vgl. Schulinternes Curriculum IF in der Sek 1 (g8 - Version 22.03.2012)

1.4 Informatikunterrichts in der Gymnasialen Oberstufe

Die Vorgaben für das Zentralabitur NRW regeln im Wesentlichen den Ablauf des Informatikunterrichts in der Gymnasialen Oberstufe.

Als Besonderheiten am Wilhelm-Hittorf-Gymnasium seien aber genannt:

- In der EF werden Schülerinnen und Schüler getrennt in den Kursen IF8 (für Schülerinnen und Schüler, die im WP-II-Bereich schon Informatikunterricht hatten) und IFneu unterrichtet. **Der hier vorgestellte schulinterne Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe ist zunächst für die Neueinsteiger (IFneu) konzipiert.**
- Als Entwicklungsumgebungen für den Einstieg in die Objektorientierte Programmierung werden Greenfoot und BlueJ/JavaEditor genutzt.
- In Zusammenarbeit mit dem Kooperationspartner guidecom in Münster werden gemeinsame Veranstaltungen (z.B. Bewerbertraining in der Q2, Softwaredesign in der EF) durchgeführt.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Unterrichtsvorhaben EF-I: Einführung in die Informatik

Bemerkung: Thema und Leitfragen sind vom Unterrichtsvorhaben EF-II nicht trennbar.#

Thema: Einführung in die Informatik

Leitfragen: Womit beschäftigt sich das Fach Informatik in der GOST? Was sind die Fragestellungen und Meilensteine der Informatik?

Vorhabenbezogene Konkretisierung:

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei sollen die Meilensteine in der Geschichte der Informatik und grundlegende Prinzipien wie das EVA-Prinzip und die Von-Neumann-Architektur erarbeitet werden.

Mit Hilfe eines geschichtlichen Abrisses und populärer informatischer Fallbeispiele sollen die Wechselbeziehung zwischen der Informatik und der Gesellschaft thematisiert und die Auswirkungen des Einsatzes von Informatiksystemen auf die Gesellschaft deutlich werden.

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.

Das gesamte Vorhaben wird durch eine eintägige Exkursion in das Heinz-Nixdorf-Forum nach Paderborn flankiert.

Zeitbedarf: 6 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>1. Geschichte und „Geschichtchen“ der Informatik und ihre Auswirkungen auf die heutige Gesellschaft</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) 	<ul style="list-style-type: none"> • Film: „Als die Computer rechnen lernten“ • Fallbeispiele (berühmte Softwarefehler: z.B. Scud-Rakete im Irak-Krieg⁴, Softwarefehler bei EC-Karten⁵, Apollo-Mondlandung⁶, Ariane 5)
<p>2. Aufbau informatischer Systeme</p> <ul style="list-style-type: none"> • Identifikation typischer Arbeitsweisen und Komponenten informatischer Systeme. • Herleitung der VN-Architektur • Identifikation des EVA-Prinzips als Prinzip der Verarbeitung von Daten und Grundlage der VN-Architektur. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der VN-Architektur (A) • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation 	<ul style="list-style-type: none"> • Aus den vorherigen Beispielen und Diskussionen werden das EVA-Prinzip die in der VN-Architektur vorkommenden Komponenten herausgearbeitet. • Hier ist die Vergabe von Referaten oder eine arbeitsteilige Erarbeitung angedacht.

⁴ u.a. http://de.wikipedia.org/wiki/MIM-104_Patriot

⁵ Informatik 1 – Schönigh – Seite 125

⁶ u.a. <http://www.bernd-leitenberger.de/apollo-ueberfluessige-fakten.shtml>

Unterrichtsvorhaben EF-II: Einführung in die Informatik – Informatisches Problemlösen und Algorithmen als Beschreibung von Abläufen

Bemerkung: Thema und Leitfragen sind vom Unterrichtsvorhaben EF-I nicht trennbar.

Thema: Einführung in die Informatik – Informatisches Problemlösen und Algorithmen als Beschreibung von Abläufen

Leitfragen: Wie beschreibt man in der Informatik Abläufe computernah? Was zeichnet Algorithmen aus und wie kann man sie vergleichen?

Vorhabenbezogene Konkretisierung:

Anhang von alltäglichen Beispielen werden zunächst der Algorithmus-Begriff und seine Eigenschaften herausgearbeitet. Dabei sollen Algorithmen sowohl umgangssprachlich wie auch im PAP dargestellt. Bedeutsam ist dabei die Herausarbeitung der grundlegenden algorithmischen Strukturen (Sequenz, bedingte Verzweigung und kopfgesteuerte Schleife) und die Überprüfung der Eigenschaften von Algorithmen.

Im Anschluss werden anhand von konkreten Algorithmen die erarbeiteten Techniken geübt, Fragestellung zu Grenzen von Algorithmen thematisiert und die Idee der rekursiv formulierten Algorithmen erarbeitet.

Beim Labyrinth-Problem führt zum ersten Mal auf Modellentscheidungen (welche Sensoren hat der Akteur? Welche Aufträge und Anfragen liegen vor?) und dient als Übergang zur ersten Implementierung in Java (Unterrichtsvorhaben EF-III).

Zeitbedarf: 9 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Der Algorithmus-Begriff	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> kennen den Begriff „Algorithmus“ und seine Eigenschaften, stellen Algorithmen umgangssprachlich und 	<ul style="list-style-type: none"> Ausgehend von den berühmten Softwarefehler⁷ und mit Hilfe alltäglicher Beispiele wird der Algorithmus-Begriff⁸ entwickelt und an

⁷ Unterrichtsvorhaben EF-I

⁸ u.a. <http://www.netzmafia.de/skripten/buecher/perl/algorithmen.pdf> (Kapitel 2, Seite 15-21)
oder http://intranet.lsg.musin.de/sj1213/images/e/e4/Ppt_EinstiegAlgorithmen_LB.pdf (Folien 1-7)

	<p>grafisch (PAP) dar,</p> <ul style="list-style-type: none"> • analysieren und erläutern einfache Algorithmen (A) 	<p>Beispielen überprüft, geschärft und eingeübt.</p> <ul style="list-style-type: none"> • Mögliche Beispiele: Papierflieger (Sequenz), Kochrezept (Sequenz und bedingte Verzweigung) ... • Material: PAPDesigner
<p>2. Beispiele für einfache Algorithmen und ihre Darstellung</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), • modifizieren einfache Algorithmen (A), • testen Programme schrittweise anhand von Beispielen (A) 	<ul style="list-style-type: none"> • Euklidischer Algorithmus⁹: Darstellung eines iterativen mathematischen Algorithmus mit Hilfe eines PAPs (Bedingte Verzweigung, kopfgesteuerte Schleife) und ugs. Beschreibung • Labyrinth-Problem: Die Rechte-Hand-Regel Darstellung in Umgangssprache und im PAP / Testen von Algorithmen / Übertragung auf vergleichbare Labyrinth / Grenze des Algorithmus (Problemklasse) • Türme von Hanoi¹⁰: Herausarbeiten von Regeln und Fragestellung aus der Geschichte, Reduzierung, praktisches Ausprobieren, Darstellung als Algorithmus, Zeitproblematik (Grenze der Umsetzung des Algorithmus) Rekursive Darstellung; Eigenschaften von Rekursionen (Selbstaufruf mit Laufbedingung bei Verkleinerung des Problems)

⁹ z.B. http://de.wikipedia.org/wiki/Euklidischer_Algorithmus

¹⁰ Simulation und Geschichte unter <http://www.blinde-kuh.de/spiele/hanoi/>
vertiefende Materialien: <http://www.mathematik.uni-muenchen.de/~hinz/turm.pdf>

Unterrichtsvorhaben EF-III: Einführung in die OOP mit Greenfoot

Thema: Einführung in die OOP mit Greenfoot

Leitfragen: Was sind Klassen, Objekte, Methoden, Attribute? Wie interagieren Objekte unterschiedlicher und gleicher Klassen miteinander? Wie werden Algorithmen entwickelt, dargestellt und in der Programmiersprache Java umgesetzt?

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beinhaltet eine Einführung in die OOP. Dabei werden Algorithmen erstellt und implementiert. In Kurzprojekten werden die Klassen und ihre Beziehungen in Form von UML-Diagrammen modelliert.

Begonnen wird mit einem Greenfoot-Szenario (i.d.R. Roboter, Ameisen, Krabben o.ä.). Nach der sukzessiven Einführung der zu behandelnden Kontrollstrukturen Verzweigung, Schleifen werden zunehmend komplexere Problemstellungen von den SuS erarbeitet und gelöst, wobei die Modellierung als Vorstufe der Implementation eingefordert wird.

In einem weiteren Schritt sollen Objekte miteinander interagieren können (z.B. Roboter und Akku, zwei Roboter). Hierbei wird der Unterschied zwischen drei Beziehungsarten (Hat, Kennt und Ist) erarbeitet und visualisiert.

Zeitbedarf: 25 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) Grundlagen der OOP/Algorithmik</p> <ul style="list-style-type: none"> • Erarbeitung der Bedeutung von Klassen, Objekten und Methoden • Darstellung von Klassen und Objekten in UML-Notation • Benutzung von Objekten • Erlernen grundlegender Algorithmen mit Kontrollstrukturen • Erarbeitung des Unterschieds zwischen Aufträgen und Abfragen anhand von Anwendungsbeispielen <p>b) Objektbeziehungen modellieren und umsetzen</p> <ul style="list-style-type: none"> • Analyse von Problemstellungen, bei denen die Interaktion von Objekten notwendig ist • Darstellung der Hat-, Kennt- und Ist-Beziehung in Form von Klassendiagrammen • Erarbeitung, Programmierung und Bewertung verschiedener Umsetzungsmöglichkeiten der Beziehungen <p>c) Einführung des Variablenkonzepts/Datentypen</p> <ul style="list-style-type: none"> • Problemorientierte Erarbeitung 	<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). • stellen den Zustand eines Objekts dar (D). • analysieren und erläutern einfache Algorithmen und Programme (A). • modifizieren einfache Algorithmen und Programme (I). • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). • analysieren und erläutern eine objektorientierte Modellierung (A) • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M). • modellieren Klassen unter Verwendung von Vererbung (M). • stellen die Kommunikation zwischen Objekten grafisch dar (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in 	<p>Greenfoot</p> <p>Szenarien für Greenfoot, z.B.:</p> <p>Wombats, Roboter, Krabben, Ameisen usw.</p> <p>PAP-Designer</p>

<p>der Benutzung von Variablen als universellem Speicher.</p> <ul style="list-style-type: none">• Benutzung der Datentypen Integer, String, Char, Boolean. <p>d) Vertiefung der Inhalte anhand eines weiteren Anwendungsbeispiels</p>	<p>Diagrammen grafisch dar (D).</p> <ul style="list-style-type: none">• analysieren und erläutern eine objektorientierte Modellierung (A).• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I).• testen Programme schrittweise anhand von Beispielen (I).• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).	
---	--	--

Unterrichtsvorhaben EF-IV: Suchen und Sortieren

Thema: Suchen und Sortieren

Leitfragen: Wie funktioniert die effektive Suche nach Objekten? Wie funktioniert das Sortieren von Objekten? Welche Algorithmen gibt es und wie effizient sind diese?

Vorhabenbezogene Konkretisierung: Anhand von alltäglichen Beispielen wird das Suchen nach Objekten motiviert. Dabei wird zunächst die iterative Suche betrachtet. Es soll deutlich werden, dass ein Objekt wesentlich schneller gefunden werden kann, wenn eine Vorsortierung der zu durchsuchenden Objekte vorliegt. Daraus soll der Algorithmus der binären Suche entwickelt werden. Die benötigte Vorsortierung motiviert die Betrachtung von Sortieralgorithmen. Als einfache Algorithmen wird hier beispielhaft der Selection- oder Bubblesort betrachtet. Dazu gehört die Modellierung, Implementierung in Greenfoot (in einem vorgefertigtem Szenario) und Untersuchung auf Effizienz, die sich auf die Anzahl der Vergleiche und Vertauschungen beschränkt. Die Algorithmen in diesem Unterrichtsvorhaben werden in Programmablaufplänen modelliert.

Zeitbedarf: 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) Suchalgorithmen (iterative und binäre Suche)</p> <p>b) Einfache Sortieralgorithmen (Selection- / Bubblesort)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D), • entwerfen einen weiteren Algorithmus zum Sortieren (M) • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf. (A) <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • analysieren und erläutern einfache Algorithmen und Programme (A), • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen 	<ul style="list-style-type: none"> • Suchen in Alltagsbeispielen (z.B.: CD-Regal, Suche der größten/kleinsten Zahl) • Betrachtung der Effizienz der iterativen Suche • Optimierung der iterativen Suche zur binären Suche. (Die Vorsortierung als Bedingung) • Beschreibung der Algorithmen in PAPs • Umsetzung des Bubblesorts oder Selectionsorts in Greenfoot (ohne Arrays) • Zählen von Zugriffen / Vergleichen / (Dreieckstausch)

Unterrichtsvorhaben EF-V: Softwareentwicklung

Thema: Softwareentwicklung

Leitfragen: Wie wird Software entwickelt? Welche Phasen durchläuft Software während der Entwicklung? Welche Bereiche werden strikt voneinander getrennt (MVC-Konzept)?

Vorhabenbezogene Konkretisierung:

Schülerinnen und Schüler durchlaufen in diesem UV einen vollständigen Softwareentwicklungsprozess am Beispiel des Spiels „Verflixte 7“.

Im Rahmen der Unterrichtsreihe erlernen sie die einzelnen Phasen OOA , OOD bis hin zur OOP. In der OOA Phase werden beteiligte Objekte mit ihren Fähigkeiten und Eigenschaften mit Hilfe der Substantiv-Verb Methode, Use-Case Diagramme und der CRC Methode identifiziert. Die OOD dient der Konkretisierung der OOA und Implementationsdiagramme werden erstellt.

Der letzte Schritt ist die Umsetzung des Projekts/ Modells mit Java einschließlich der Gestaltung einer grafischen Oberfläche. Sequenzdiagramme helfen das System zu testen und bei Bedarf zu verbessern.

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) Exkurs binäre Zahlen Im Rahmen eines kurzen Exkurses werden Fragestellungen rund um die Binärdarstellung besprochen.</p> <p>b) OOA Die OOA dient der Untersuchung der Aufgabenstellung und führt zu einem ersten UML-Modell. Die SuS benutzen dazu die Fachmethoden: USE-Case Diagramme, Substantiv Verb Methode, CRC Karten</p> <p>c) OOD Die OOD dient der Konkretisierung und führt zu einem Implementationsdiagramm</p> <p>d) OOP Mit BlueJ wird das Modell implementiert und unter anderem mit Hilfe des Debuggers und von Sequenzdiagrammen auf seine Lauffähigkeit getestet.</p> <p>e) MVC Konzept Das Modell wird mit Hilfe des Java-Editors um eine einfache grafische Oberfläche erweitert.</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D). • 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), • modellieren Klassen unter Verwendung von Vererbung (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), • stellen den Zustand eines Objekts dar (D), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • implementieren Klassen in einer Programmiersprache (I). • analysieren und erläutern 	<ul style="list-style-type: none"> • Use Case Diagramme • Substantiv-Verb Methode • CRC Karten • Sequenzdiagramme • BlueJ • Java Editor

	<p>einfache Algorithmen und Programme (A),</p> <ul style="list-style-type: none">• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).• analysieren und erläutern eine objektorientierte Modellierung (A)	
--	---	--

2.2 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

2.2.1 Vorbemerkungen

Die rechtlich verbindlichen Vorgaben zur Leistungsbewertung finden sich in § 48 SchulG, in § 6 der APO-SI sowie den §§ 13-17 der APO-GOST. Danach ist eine rein rechnerische Bildung von Abschlussnoten unzulässig. Die Fachkonferenz legt nach § 70 SchulG Grundsätze zu Verfahren und Kriterien der Leistungsbewertung fest. Sie orientiert sich dabei an den in den Lehrplänen für die Sekundarstufen I und II ausgewiesenen Kompetenzen.

Das fachbezogene Leistungskonzept ist für alle Mitglieder einer Fachschaft verbindlich. Es soll für ein möglichst hohes Maß an Transparenz und Vergleichbarkeit von Leistungsbeurteilungen sorgen. Rückfragen zum Leistungsstand richten Schülerinnen und Schüler sowie Eltern bitte immer zunächst an die unterrichtenden Fachlehrerinnen und Fachlehrer.

Die Grundsätze der Leistungsbewertung werden den Schülerinnen und Schülern immer zum Schuljahresbeginn, bei Lehrerwechsel auch zum Halbjahresbeginn mitgeteilt. Ein Hinweis darauf wird im Klassenbuch / Kursheft vermerkt. Die Erziehungsberechtigten werden im Rahmen der Elternmitwirkung von den Klassenlehrerinnen und Klassenlehrern auf der ersten Klassenpflegschaftssitzung informiert und darauf hingewiesen, dass die Leistungskonzepte der einzelnen Fächer auf der Homepage des W-H-G zu finden sind.

Kriterien der Leistungsbewertung im Zusammenhang mit konkreten, insbesondere offenen Arbeitsformen werden den Schülerinnen und Schülern grundsätzlich vor deren Beginn transparent gemacht.

Jede Lehrerin / jeder Lehrer dokumentiert regelmäßig und kontinuierlich die von den Schülerinnen und Schülern erbrachten Leistungen.

Die Leistungsrückmeldung erfolgt in regelmäßigen Abständen (zumindest zum Quartalsende) differenziert und individuell in schriftlicher oder mündlicher Form.

Bei Minderleistungen erhalten die Schülerinnen und Schüler sowie ihre Erziehungsberechtigten im Zusammenhang mit den Halbjahreszeugnissen individuelle Lern- und Förderempfehlungen, die die Lernenden - ihrem jeweiligen Leistungsstand entsprechend - zum Weiterlernen ermutigen, indem sie Hinweise zu Erfolg versprechenden individuellen Lernstrategien geben. Den Eltern werden im Rahmen der Lern- und Förderempfehlungen Wege aufgezeigt, wie sie das Lernen ihrer Kinder unterstützen können.

Bei Elternsprechtagen und im Rahmen regelmäßiger Sprechstunden erhalten die Erziehungsberechtigten Gelegenheit, sich über den Leistungsstand ihrer Kinder zu informieren und dabei Perspektiven für die weitere Lernentwicklung zu besprechen.

2.2.2 Bewertungsbereich Sonstige Mitarbeit

Die Beurteilung der sonstigen Mitarbeit erfolgt gemäß dem Lehrplan Informatik. Sie erfasst die Qualität, die Quantität und die Kontinuität verschiedener Beiträge. Für die Bewertung der Leistungen sind sowohl Inhalts- als auch Darstellungsleistungen zu berücksichtigen. Die sonstige Mitarbeit wird dabei in einem kontinuierlichen Prozess während des Schulhalbjahres festgestellt.

Formen der sonstigen Mitarbeit

- Mündliche Beteiligung am Unterrichtsgespräch
 - o Regelmäßige Beteiligung (quantitativ und qualitativ)
 - o Wiedergabe von Kenntnissen und Prozessen
 - o Darstellung von Problemsituationen
 - o Beiträge zur Entwicklung von Problemlösungen und Bewertung von Arbeitsständen und –ergebnissen
 - o Vorstellung von Arbeitsergebnissen aus Gruppen- oder Partnerarbeit
 - o Nutzung adäquater Fachsprache
- Selbstständige Arbeit zu zweit oder in Gruppen.
 - o Mitarbeit bei der Organisation (Absprachen, Arbeitseinteilung, Schnittstellenvereinbarung, gegenseitige Hilfen) der gemeinsamen Arbeit
 - o Kooperatives Verhalten und gemeinsame konstruktive Suche nach Lösungswegen bei den gestellten Problemen
 - o Erreichen des Arbeitszieles unter Einsatz der zur Verfügung stehenden Materialien und der erlernten Arbeitstechniken
- Praktische Arbeit am Computer
 - o Vor- und Nachbereitung der Rechnerarbeitsphasen (u.a. Dokumentationen)
 - o Kooperatives Arbeiten am Rechner und Beratung anderer Arbeitsgruppen
 - o Sinnstiftender Umgang mit den vorhandenen Ressourcen (Nutzung von Hardware, Werkzeugen, insbesondere Umgang mit der Programmierumgebung, angemessene Reaktion auf Fehlermeldungen usw.)
- Anfertigung und Präsentation von Hausaufgaben
 - o Vor- und Nachbereitung von Unterrichtsstunden
 - o Intensive und möglichst selbstständige Auseinandersetzung mit Materialien und schriftlichen Informationen

- Selbstständige Weiterarbeit und Vervollständigung von Programmierlösungen (auch mit Hilfe des SLZ).
- Bis zu eine schriftlichen Übung pro Halbjahr
- Evtl. Erstellen von Protokollen
- Evtl. Referate

Die Reihenfolge der oben angegebenen Punkte spiegelt die Bedeutung bei der Beurteilung im Sinne einer Rangfolge wider, wobei nach konkretem Unterrichtsverlauf die Gewichtungen abweichen können.

2.2.3 Klausuren

Es ist Beschluss der Fachschaft, im 1. Halbjahr der Einführungsphase nur eine Klausur (2. Quartal) zu schreiben. Im zweiten Halbjahr der Einführungsphase kann eine Klausur durch eine Projektarbeit ersetzt werden. Die Dauer der Klausuren ist laut APO-GOST vorgegeben.

Kompetenzen: Analyse, Algorithmik, Programmiertechnik, Modellierung, Umgang mit den vorhandenen Ressourcen

Materialgrundlage: vorhandene Ressourcen

Aufgabenformate: Reproduktive Aufgabenteile, Modellierung und Realisierung von Programmen

zugelassene Hilfsmittel: keine

Übersicht über Gewichtung/Punkteverteilung - Bewertungskriterien

Jede Aufgabe wird nach Bedeutung der überprüften Lernziele, Anteil der erwarteten Arbeitszeit, Umfang und Komplexität der einzubringenden Teilleistungen, Grad der geforderten Selbstständigkeit und Art und Form der Darstellung entsprechend bepunktet und dieses den Schülerinnen und Schülern offengelegt. Der Hilfscharakter eines solchen Bewertungsverfahrens ist zu betonen und die Gesamtnote ist nicht unbedingt und unflexibel aus der erreichten Prozentzahl zu bestimmen.

Notentabelle:

Durch Leistungen mit vorwiegend wiederholendem Charakter ist eine ausreichende Bewertung erzielen zu können. In der Regel reichen 40% der Maximalpunktzahl zu einem ausreichenden Ergebnis. Die Notenstufen ergeben sich i.a. daraus aus äquidistanter Einteilung.

Die Dokumentation der Leistungsbewertung enthält Kriterienraster und Förderhinweise.

3 Weitere Hinweise und Entscheidungen

Innerhalb der EF soll eine Exkursion zum Heinz-Nixdorf-Forum nach Paderborn unternommen werden, bei dem die im Lehrplan thematisierten Bereiche aus der Geschichte der Informatik und dem Bereich „Informatik, Mensch und Gesellschaft“ aufgegriffen werden.

Innerhalb der EF soll im Rahmen der Softwareentwicklung eine Veranstaltung mit der guidecom gemacht werden.

Die Fachschaft Informatik strebt an, das Buch „Informatik 1“ aus dem Schöningh-Verlag nach Erscheinen der Neuauflage (ca. Mai 2014) als Lehrbuch für die EF einzuführen.

4 Qualitätssicherung und Evaluation

4.1. Kollegialer Austausch

Durch einen regelmäßigen Austausch innerhalb der Fachschaft im Hinblick auf unterrichtlich eingesetzte Materialien und Methoden wird eine Homogenisierung des Unterrichts angestrebt. Zudem ist der Austausch von gestellten Klausuren obligatorisch.

Ein kollegialer Austausch findet auch zwischen unserer Fachschaft und der Informatik-Fachschaft des kooperierenden Annette-v-Droste-Hülshoff-Gymnasiums statt.

4.2. Evaluation durch die Schülerinnen und Schüler

Eine Evaluation hinsichtlich in der Fachgruppe strittig diskutierter Entscheidungen im neuen schulinternen Lehrplan soll den Diskussions- und Entscheidungsprozess in der Fachschaft zum Ende des Sj 2014/2015 unterstützen.

Entsprechend sollen die Leistungen der Schülerinnen und Schüler in der Qualifikationsphase kritisch auf die Planungsentscheidungen der EF reflektiert werden.

5 Kompetenzerwartung und inhaltliche Schwerpunkte in der Einführungsphase

Inhaltsfeld 1: Daten und ihr Strukturierung

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
Die SuS ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)	Die SuS können anhand alltäglicher Beispiele Klassen identifizieren, wesentliche Eigenschaften und Fähigkeiten benennen, sowie Aufträge und Anfragen mit Art des Rückgabewerts unterscheiden und die Klasse grafisch darstellen. Sie unterscheiden zwischen Klassen und konkreten Exemplaren.	Bei der Identifizierung der Rückgabewerte genügt die Einteilung in Wahrheitswert, Zahl und Text. Eine Unterscheidung zwischen Ganzzahl und Kommazahl wird nicht erwartet. Als Darstellungsform wird das UML-Diagramm genutzt. Fachsprachlich wird die Vorstellung von Klassen als Baupläne und von Objekten als Exemplare vermittelt, wobei das Wort „Objekt“ aufgrund seiner Doppeldeutigkeit zunächst vermieden wird.	Alltägliche Beispiele können Schüler (Mitschüler), Lehrer, Klassenräume, Tiere ...sein.
modellieren Klassen mit ihren Attributen, ihren Methoden und Assozia-tionsbeziehungen (M)			
modellieren Klassen unter Verwendung von Vererbung (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)			

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
stellen den Zustand eines Objekts dar (D)			
stellen die Kommunikation zwischen Objekten grafisch dar (M)			
stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D)			
dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)			
analysieren und erläutern eine objektorientierte Modellierung (A)			
implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).			

Inhaltsfeld 2: Algorithmen

Inhaltliche Schwerpunkte:

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

Analyse, Entwurf und Implementierung einfacher Algorithmen:

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
Die SuS analysieren und erläutern einfache Algorithmen und Programme (A)	Die SuS können im Rahmen des Programmierprojekts Methoden mit einfachen Kontrollstrukturen verstehen und die Funktionsweise erklären.	Verzweigungen und Schleifen werden von den SuS erkannt und in der Wirkung beschrieben. Als Darstellungsform wird das PAP benutzt.	Problemorientierte Beispiele wie z.B. Mauern oder das Ende der Spielwelt des betrachteten Szenarios dienen als Gegenstand zur Erarbeitung. Werkzeuge sind Greenfoot, BlueJ, StruktED und PAP-Designer.
Die SuS modifizieren einfache Algorithmen und Programme (I),	Die SuS können Methoden je nach Aufgabe verändern und erweitern.	Bedingungen bei den Verzweigungen und Abbruchbedingungen bei Schleifen werden so verändert, dass weiterführende Aufgaben gelöst werden. Die Manipulation von Attributen auch durch Parameter wird ebenfalls benutzt.	Z.B. das Ablaufen eines Labyrinths wird in verschiedenen Schwierigkeitsgraden entwickelt. In einem weiterführenden Projekt werden die algorithmischen Fachkenntnisse vertieft.
Die SuS entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),	Die SuS können eindeutige Handlungsvorschriften umgangssprachlich beschreiben und als Pseudocode formulieren. Abläufe werden durch geeignete Diagramme	Programmabläufe, insbesondere Schleifen und Verzweigungen werden in Form von Struktogrammen dargestellt und als Pseudocode formuliert.	Werkzeug ist der PAP-Designer.

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
	dargestellt		
Die SuS implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),	Die SuS können Algorithmen im Methodenkonzept programmieren. Dabei werden Variablen verändert und Parameter genutzt. Einfache Kontrollstrukturen werden von den SuS beherrscht.	Die Anwendung des Variablenkonzepts wird von den SuS beherrscht und bei der Umsetzung von Algorithmen angewendet. Verzweigungen sowie Schleifen werden auch in komplexeren Zusammenhängen geschachtelt benutzt.	Werkzeuge sind Greenfoot, BlueJ und der Java-Editor
Die SuS testen Programme schrittweise anhand von Beispielen (I).	Die SuS erschließen sich die Funktionsweise von Methoden systematisch in verschiedenen Problemsituationen. Sie können durch gezielte Wertebelegung von Attributen den Ablauf nachvollziehen.	Ausgewählte oder von den SuS selbst implementierte Programme/Methoden können sowohl auf Fehler, als auch auf ihre Funktionsweise hin untersucht werden.	Werkzeuge: BlueJ-Debugger
Die Schülerinnen und Schüler analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D),	Die SuS kennen den Unterschied zwischen Linearer- und Binärer Suche und können grundlegende Beispielverfahren anwenden. Sie können die Effektivität einschätzen.	Ein grundlegendes Such- und Sortierverfahren wird besprochen und erprobt. Anhand von Beispielszenarien wird die Effektivität untersucht.	Lebensnahe Kontexte wie z.B. Schülerlisten, Binäre Suche in großen Datenmengen dienen als Gegenstand Sortierverfahren: Bubblesort).
Die SuS entwerfen einen weiteren Algorithmus zum Sortieren (M),	Auf der Basis eines Basisalgorithmus wird ein weiteres Verfahren entwickelt.	Die SuS vertiefen ihr Verständnis bzgl. Sortiervorgängen, indem z.B. die Trennung von sortiertem- und unsortiertem Bereich thematisiert und in einem eigenen Verfahren angewendet wird.	Gearbeitet wird begrenzten Zahlenräumen. Es genügt die theoretische Erarbeitung der Such- und Sortierverfahren.

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
<p>Die SuS beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A).</p>	<p>Die SuS können anhand von Grundlegenden Sortierverfahren deren Effektivität beurteilen. Dabei können Sie auf der Basis des Speicherplatzbedarf und der Zahl der Operationen argumentieren. Sie können beurteilen, ob ein Verfahren mit dem ursprünglichen Speicherplatz auskommt, oder zusätzlichen benötigt.</p>	<p>Die SuS untersuchen die Zahl der Operationen (Vergleiche, Vertauschungen). Beispielhaft wird ein Verfahren betrachtet, das nicht In-Place arbeitet. Aufwand wird anhand einer Tabelle beurteilt.</p>	<p>Bubblesort (und ein weiteres Verfahren mit quadratischem Aufwand). Tabellendarstellung der Anzahl der Operationen.</p>

Inhaltsfeld 3: Formale Sprachen und Automaten

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
Die SuS implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),	Die SuS können Quelltext in korrekter Syntax und Semantik programmieren. Typische Fehlermeldungen werden von den SuS richtig eingeschätzt. Fehler im Programm können weitgehend eigenständig beseitigt werden.	Die richtige Syntax der Sprache wird kontinuierlich im Unterricht aufgegriffen und geübt.	Durch die übersichtliche Strukturierung von Quelltext wird die fehlerarme Programmierung gefördert. Die farbliche Hervorhebung von Quelltextelementen in BlueJ wird benutzt.
interpretieren Fehlermeldungen und korrigieren den Quellcode (I)	Die SuS können Fehlermeldungen richtig einschätzen und dazu benutzen, um Fehler im Quelltext gezielt zu beseitigen.	Mit Hilfe von Fehlermeldung und Debugger werden Fehler im Programm systematisch behoben. Typische Fehlermeldungen werden im Unterricht thematisiert.	Häufige Fehler wie Klammerung, Groß- und Kleinschreibung werden unterrichtsbegleitend thematisiert. Als Hilfsmittel wird der BlueJ-Debugger benutzt.

Inhaltsfeld 4: Informatiksysteme

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
<p><i>Digitalisierung</i> Die SuS stellen ganze Zahlen und Zeichen in Binärcodes dar (D). Die SuS interpretieren Binärcodes als Zahlen und Zeichen (D).</p>	<p>Die SuS wissen, dass die Verarbeitung und die Speicherung von Daten letztlich immer auf zwei Zustände basieren. Die SuS begründen den Zahlüberlauf bei Integer-Zahlen anhand der binären Zahldarstellung im Rechner. Die SuS können Buchstaben mit Hilfe der AscII-Code-Tabelle in dezimale und binäre Zahlen umsetzen und so Buchstaben im Alphabet unter Berücksichtigung von Wertebereichen verschieben.</p>	<p>SuS aus dem IF8-Kurs kennen die Binärdarstellung aus dem WP11-Bereich (logische Schaltungen). SuS aus dem IF_EF-Kurs erarbeiten in einem Exkurs Fragestellungen rund um die Binärdarstellung, die Speicherung von Daten und der logischen Verknüpfung von Aussagen.</p>	<p>Im Zusammenhang mit der ASCII-Code-Tabelle und der binären Zahldarstellung (ggf. Bitmapdarstellung) wird deutlich, dass alle Darstellungsweisen binär reduzierbar sind. Material aus „Informatik 2“ – Seite 38-45 Im Kontext von Caesar-Verschlüsselung werden diese Erkenntnisse programmiertechnisch umgesetzt.</p>
<p><i>Einzelrechner</i> Die SuS beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).</p>	<p>Die SuS kennen den Grundaufbau eines VN-R. und können die entscheidende Idee nennen.</p>	<p>Die entscheidende Idee des VN-R. (Programm wird behandelt wie Daten und damit wird der Rechner zur universellen Maschine) und das EVA-Prinzip werden herausgearbeitet. Eine geschichtliche Einordnung der Idee geschieht. Bearbeitung dieses Aspektes ggf. als Schülerreferat.</p>	<p>Als Material wird das Leitprogramm der RWTH Aachen zum VN-R. (Kapitel 1) und das Informatik-Duden-Buch genutzt.</p>
<p><i>Dateisystem</i> Die SuS nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen</p>	<p>Die SuS können ihren Dateien sinnvoll benennen, in Ordner sinnvoll organisiert ablegen.</p>	<p>Es soll im Zusammenhang mit den .java- und .class-Dateien die Begriffe Quellcode und Bytecode thematisiert und damit die</p>	<p>Auf dem Laufwerk N wird für jeden Kurs ein gemeinsamer Ordner eingerichtet.</p>

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).	Sie können die verschiedenen Dateitypen, die in einem Java-Projekt (Greenfoot und BlueJ) auftreten unterscheiden.	Plattformunabhängigkeit von Java hinterfragt werden.	
<i>Internet</i> Die Schülerinnen und Schüler nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	Die SuS nutzen Programmiererergebnisse ihrer Mitschülerinnen und –schüler und des Lehrers und aus weiteren Quellen und stellen auch eigenen Quellcode anderen bereit. Übernommener oder zur Verfügung gestellter Programmcode soll adäquat debattiert werden.	Die Übernahme von Quellcode anderer Mits, des Lehrers und aus anderen Quellen wird thematisiert und in diesem Zusammenhang der Schutz des geistigen Eigentums besprochen.	

Inhaltsfeld 5: Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- ☒ Einsatz von Informatiksystemen
- ☒ Wirkungen der Automatisierung
- ☒ Geschichte der automatischen Datenverarbeitung

Inhaltsfelder (Gegenstände) Kompetenzbereiche (Prozesse)	Kompetenzerwartungen	Didaktische Hinweise und Vereinbarungen	Hinweise und Vereinbarungen zu geeigneten Kontexten
Einsatz von Informatiksysteme: Die Schülerinnen und Schüler nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).	Die SuS können die zur Verfügung gestellte Hardware und die für den Unterricht eingeführte Software sicher und reflektiert einsetzen.		In der EF werden folgende Werkzeuge eingesetzt: - Greenfoot - BlueJ - PAPDesigner - Java-Editor
Wirkungen der Automatisierung Die Schülerinnen und Schüler bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).			- VN-Architektur - Ggf. Auswirkung der Kryptologie im 2. Weltkrieg auf die Entwicklung der Informatik
Geschichte der automatischen Datenverarbeitung Die Schülerinnen und Schüler erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).	Die SuS kennen die Meilensteine der automatisierten Datenverarbeitung und die dabei auftretenden entscheidenden Ideen.		