
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 CRG hat Themen, Leitfragen und die Ausführungen unter der Überschrift *Vorhabenbezogene Konkretisierung* verbindlich vereinbart, ebenso die zur Sequenzierung der Unterrichtsvorhaben ausgewiesenen Inhalte und Kompetenzen (erste und zweite Tabellenspalte). Alle Mitglieder der Fachkonferenz haben sich darauf verständigt, in ihrem Unterricht Lerngelegenheiten anzubieten, so dass Schülerinnen und Schüler diese Kompetenzen im Rahmen ihrer Unterrichtssequenz erwerben oder vertiefen können.

Unterrichtliche Anregungen:

Die angeführten Beispielkontexte, Medien und Materialien sind dagegen als Vorschläge zu verstehen. Hier werden einige bewährte Projekte und Materialien aufgeführt, mit denen sich die Sequenzziele der ersten beiden Spalten umsetzen lassen.

Insbesondere wird hier auf Kapitel des eingeführten Lehrwerks (B. Schriek: Informatik mit Java) hingewiesen. Das Lehrbuch für die EF (Band 1) wird im folgenden mit *Schriek* abgekürzt.

Die Verwendung alternativer Kontexte, Materialien und Projekte ist jederzeit möglich.

I) Einführungsphase

Die folgenden Kompetenzen 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),
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).

Unterrichtsvorhaben EF-I

Thema: *Einführung in Aufbau und Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten bis zu ersten Javaprogrammen*

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Wie ist ein digitaler Rechner prinzipiell aufgebaut? Wie liegen Daten im Speicher vor und wie kann man auf diese Daten zugreifen und sie bearbeiten? Was ist ein Programm und wofür benötigt man Programmiersprachen?*

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. Ein prinzipielles Grundverständnis der Abläufe in einem Digitalrechner erleichtert außerdem später das Verständnis bei der in Java notwendigen Deklaration aller Variablen über die Verknüpfung eines Variablenbezeichners mit einem Datentyp und beim grundlegenden Befehl der Zuweisung eines Wertes an eine Variable.

Ausgangspunkt könnte der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur sein, ferner der Begriff der Information bzw. deren Kodierung in Form von Daten im Speicher. Da Daten im Computer nur in binärer Form vorliegen, schließt das eine Einführung des binären Stellenwertsystems ein, welches nach Möglichkeit durch das hexadezimale Zahlensystem ergänzt und kontrastiert werden sollte.

Die Daten liegen auch auf Speichermedien nur als ganze Zahlen 0 oder 1 vor, werden uns aber durch das Betriebssystem als Dateien unterschiedlichen Typs angeboten. Erst im Zusammenhang mit der passenden Software erhalten die Daten einen Sinn und können „verarbeitet“ werden. Dabei lässt sich die Arbeit jedes *Programms* im Sinne von *Eingabe-Verarbeitung-Ausgabe* verstehen. Das Programm arbeitet dabei oft mit unterschiedlichen *Datentypen*.

Der Prozessor findet seine Daten immer nur über eine Adresse (im Prinzip eine große digitale Zahl) im Speicher bzw. auf Speichermedien wieder. So wie wir als Benutzer Dateien im Dateisystem nicht über diese Adresse sondern über ihren Dateibezeichner wiederfinden, hat es sich bewährt, dass Speicherstellen für Daten in selbst geschriebenen Programmen auch über einen wählbaren *Bezeichner* angesprochen wird, ähnlich einer Variablen in der Mathematik. Ein *Übersetzer (Compiler)* ist nötig, um für Menschen lesbare Bezeichner und Befehle einer *Programmiersprache* in die *Maschinensprache* des Prozessors zu übersetzen.

Ähnliche Übersetzungsschritte sind auch im Internet notwendig, wo binär bzw. hexadezimal vorliegende IP-Adressen durch den Namensdienst mit Bezeichnern wie www.crghagen.de verbunden werden. Das Dateisystem kann sich in vernetzten Systemen über viele Rechner erstrecken, als Beispiel wird die Ordnerstruktur und Speicherung auf dem Schulserver des CRG besprochen.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Inhalte	Zu entwickelnde Kompetenzen	Beispielkontexte, Medien, Materialien
---------	-----------------------------	---------------------------------------

<p>1. Aufbau informatischer Systeme</p> <p>(a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“</p> <p>(b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten</p> <p>2. 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 in 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), • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), 	<p><i>Material:</i> Demonstrationshardware Durch Demontage eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. Als Demonstrationsrechner bietet sich ein ausrangierter Schulrechner an.</p> <p>Betrachtung unterschiedlicher Dateien mit einem Editor, zunächst ohne Endung, so dass der Dateityp für das Betriebssystem nicht erkennbar ist.</p> <p><i>Beispiel:</i> Textkodierung Kodierung und Dekodierung von Texten mit unbekanntem Zeichensätzen (z.B. Wingdings)</p> <p><i>Beispiel:</i> Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken</p>
---	--	--

<p>3. Java als ein Beispiel für eine Programmiersprache</p> <ul style="list-style-type: none"> (a) Maschinensprache und Übersetzer, (b) BlueJ als Programmierumgebung, (c) Variablendeklaration und einfache Datentypen in Java (int, double), (d) Bezeichnerregeln (e) Wertzuweisungsbefehl, (f) evtl.: Deklaration von Parametern 		<p>Variablen als „Behälter“ für Werte visualisieren</p> <p>Einfache Geradeausprogramme mit Standardausgabe und Deklaration und Wertzuweisung im Quelltext, beispielsweise: Währungsrechner, ganzzahlige Division, pq-Formel ohne Fallunterscheidung</p> <p>Erweiterung: implizite Wertzuweisung beim Aufruf einer (<i>static</i>-)Funktion mit Parametern in BlueJ</p>
--	--	--

Unterrichtsvorhaben EF-II

Thema: *Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand einiger exemplarischen Klassen und Methoden ohne Verwendung von Kontrollstrukturen*

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch mit Hilfe von Klassen 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 und Modellierung sowie der Implementierung in der Sprache Java 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 eingeführt: Eigenschaften der Objekte führen zu *Attributen* (Instanzvariablen) der *Klasse*, Objekte sind *Exemplare (Instanzen)* einer Klasse mit jeweils eigenen Attributwerten.

Die Tätigkeiten oder Handlungen, die ein Objekt ausführt, werden abgebildet auf die *Dienste (Methoden)* der jeweiligen Klasse. Dabei werden unter den Diensten prozedurartige *Aufträge* von funktionsartigen *Anfragen*, die Werte zurückgeben unterschieden.

Relativ bald soll die textuelle Umsetzung in eigenen Java-Klassen neben die Verwendung vorgegebener Klassen aus didaktischen Bibliotheken treten, so dass der grundsätzliche Aufbau eines objektorientierten Programms in Java - welches immer aus mindestens einer Klasse besteht - deutlich wird. Die von BlueJ bereitgestellten Möglichkeiten (insbesondere der Objektinspektor) werden genutzt, um eine Vorstellung vom Aufbau eines Objektes als Einheit aus den Attributwerten und den Methoden zu entwickeln, die in unterschiedlicher Art grafisch dargestellt werden kann.

Die grundlegenden Programmbefehle *Objekterzeugung (Instanzierung)* und der *Methodenaufruf* erlauben schon einige längere Programme, insbesondere bei Verwendung von Bibliotheksklassen für Grafikelemente (SuM oder GLOOP). 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. Typische Fehler aufgrund fehlender oder falscher Deklaration oder unterlassener Objekterzeugung sollten hier mit den entsprechenden Compilermeldungen ausführlich besprochen werden.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Inhalte	Zu entwickelnde Kompetenzen	Beispielkontexte, Medien, Materialien
----------------	------------------------------------	--

<p>1. Objekte und Klassen</p> <ul style="list-style-type: none"> • Betrachtung einiger Beispiele aus dem Alltag • Unterscheidung von individuell unterschiedlichen Daten (Attributen) und klassentypischen Operationen (Methoden) • die Klasse als Bauplan und Fabrik für Objekte • der Konstruktor-Dienst • Klassendiagramme • Klassenbezeichner 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren einfache 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> Vogelschwarm <i>Beispiel:</i> Bankkonto</p> <p>verschiedene Bildschirmobjekte aus sum.kern mit BlueJ erstellen</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator - Allgemeine Objektorientierung (Download EF-II.1)</p>
<p>2. Verwendung von Klassen didaktischer Lernumgebungen neben eigenen Klassen</p> <ul style="list-style-type: none"> • Importieren von Bibliotheksklassen • Deklaration und Erzeugung von Objekten • Javatypische Klassendokumentation lesen (z.B. zu Klassen aus sum.kern) • Eigene Klassen mit javadoc dokumentieren • Methoden mit der Punktnotation aufrufen • Methodenaufrufe mit Parametern zur Übergabe von Daten an eine Objektmethode oder auch zur Änderung eines Attributwertes 		<p><i>Beispiel:</i> Die Basisobjekte Bildschirm, Maus, Tastatur und (Bunt-)Stift aus der Bibliothek sum.kern. Hier konzentriert man sich sinnvollerweise zunächst auf Bildschirm- und Stiftobjekte.</p> <p><i>Materialien:</i> Dokumentation der didaktischen Bibliothek GLOOP (Download EF-II.2)</p>

Unterrichtsvorhaben EF-III

Thema: *Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen (insbesondere Kontrollstrukturen) in Java*

Leitfragen: *Wie lassen sich Programmabläufe mit Verzweigungen oder Wiederholungen oder Abläufe in Abhängigkeit von Maus- oder Tastatureingaben in Java programmieren?*

Vorhabenbezogene Konkretisierung:

Die bisher verwendeten Sprachmittel sind erkennbar in vieler Hinsicht zu beschränkt. So könnte schon bei einem Programm zur pq-Formel deutlich werden, dass manchmal Fallunterscheidungen nötig sind, die zu unterschiedlichen Programmausgaben führen. Erst recht gilt das für Programme, die in unterschiedlicher Weise auf Benutzereingaben über Maus oder Tastatur reagieren sollen. Und bei interessanteren Grafiken stößt man leicht auf die Frage, ob man Wiederholungen nicht auch effizienter beschreiben kann als durch x-fache Wiederholung derselben Zeichenbefehle.

Sind an einem Beispiel Schleifen und Verzweigungen eingeführt worden, sollen diese Konzepte an weiteren Beispielprojekten eingeübt werden. Dabei muss es sich nicht zwangsläufig nur um solche handeln, bei denen Kontrollstrukturen zur Animation oder zu komplexeren Zeichnungen verwendet werden. Auch die Erzeugung größerer Mengen bestimmter Objekte und deren Verwaltung in einem Feld, oder komplexere Berechnungen (z.B. Primzahluche o.ä.) können hier schon Anlass zur Verwendung von Kontrollstrukturen sein.

Die Darstellung von Kontrollstrukturen in grafischer Form (als Ablaufdiagramm oder als Struktogramm) ebenso wie die Beschreibung eines Algorithmus mit Pseudocode sollte besonders geübt werden, ebenso die Analyse eines vorgegebenen Algorithmus z.B. mit Hilfe einer Ablaufabelle („Schreibtischtest“).

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

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Inhalte	Zu entwickelnde Kompetenzen	Beispielkontexte, Medien, Materialien
<p>1. Verzweigungen und Schleifen Zunächst werden die wichtigsten Kontrollstrukturen isoliert an Beispielen eingeführt, wo die Notwendigkeit dieser Konstrukte sich aus der Problemstellung ergibt</p> <ul style="list-style-type: none"> • einseitige und mehrseitige Verweigungen • Schleifen mit Eingangs- und Ausgangsbedingung • Zählschleifen • Grafische Darstellungsformen, z.B. Struktogramme 	<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), 	<p>pq-Formel erfordert Verzweigungen Zählschleifen sind nützlich um mehrfach dasselbe zu zeichnen</p> <p><i>Schriek</i> Kapitel 4: Zeichnungen, die durch Maus- oder Tastaturabfragen gesteuert werden erfordern Kontrollstrukturen</p>
<p>2. Verschachtelte Kontrollstrukturen Komplexere Algorithmen entstehen durch Kombination der Kontrollstrukturen.</p> <ul style="list-style-type: none"> • Einrückungen im Programm und im Pseudocode • Struktogramm und Schreibtischtest zum Nachvollzug eines Algorithmus • Fehler suchen und korrigieren 		<p>Freihandzeichnen mit Stift und Maus (<i>Schriek</i>, Kapitel 4)</p> <p>Computergrafik (ähnlich <i>Turtlegrafik</i> in Logo), mit verschachtelten Zählschleifen</p> <p>Komplexere Berechnungen: Primzahlen, vollkommene Zahlen usw.</p> <p><i>Schriek</i>, Kap. 5: Darts-Projekt</p>

<p>3. Datenfelder</p> <ul style="list-style-type: none"> • Deklaration • Erzeugung • Initialisierung • Durchlaufen eines Feldes 	<p>(M),</p> <ul style="list-style-type: none"> • 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), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<p>Datenfelder können in rein rechnerischen Problemstellungen eingeführt werden, oder auch genutzt werden als Liste von Daten für eine grafische Ausgabe (z.B. Koordinaten eines Zeichen-Weges) Später stehen sie dann im Vorhaben E-IV schon zur Verfügung, um auch Mengen von grafischen Objekten zu verwalten</p>
--	--	--

Unterrichtsvorhaben EF-IV

Thema: *Modellierung und Implementierung von Klassen- und Objektbeziehungen im Kontext grafisch repräsentierbarer Objekte*

Leitfrage: *Wie lassen sich unterschiedliche Beziehungen zwischen Objekten bzw. Klassen beschreiben und in Java realisieren?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich im Schwerpunkt mit dem Aufbau komplexerer Objektbeziehungen. In den vorangegangenen Unterrichtsvorhaben schickten die Objekte immer nur Nachrichten an solche Objekte, die sie vorher selbst erstellt hatten. Diese Objektbeziehung wird in diesem Vorhaben als „Hat-Beziehung“ bezeichnet werden. Dazu tritt jetzt die weniger enge „Kennt-Beziehung“, wo beliebigen Objekten, insbesondere auch solchen, die in anderen Klassen erstellt wurden, Nachrichten geschickt werden können.

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. Typische Beispiele zur Einführung solcher „Kennt-Beziehungen“ sind animierte grafische Objekte, die bei Kollisionen (mit dem Rand oder anderen bewegten Objekten) die Kollisionspartner kennen müssen.

Als dritte Beziehung zwischen Objekten wird die in Java schon auf der Klassenebene implementierte Vererbung besprochen, die hier „Ist-Beziehung“ genannt wird.. Dazu werden verschiedene Varianten der Vererbung vorgestellt, zunächst Vererbung als Spezialisierung, im Sinne einer einfachen Erweiterung einer Oberklasse, später auch Vererbung als Generalisierung mit dem Aspekt der späten Bindung, indem Dienste einer Oberklasse polymorph überschrieben werden. Dabei kann auch schon auf das Prinzip der abstrakten Klasse eingegangen werden. Dieser Inhalt ist aber nicht obligatorisch für die Einführungsphase.

Modellierungen sollen in Form von Beziehungsdiagrammen dargestellt und bei der Implementation genutzt werden.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Inhalte	Zu entwickelnde Kompetenzen	Beispielkontexte, Medien, Materialien
<p>1. Hat-Beziehung</p> <ul style="list-style-type: none"> am Beispiel von Klassen für grafisch repräsentierte Objekte Aufbau des Konstruktors einer Klasse (genauere Betrachtung als im Vorhaben EFII) <p>2. Kennt-Beziehung</p> <ul style="list-style-type: none"> Besonderheiten von Objektvariablen (Referenzen) verschiedene Möglichkeiten, eine Kennt-Beziehung herzustellen, 	<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 	<p><i>Schriek:</i> Kap. 6: Billardsimulation erweitert durch ein Feld für die Darstellung vieler Kugeln</p> <p><i>Schriek:</i> Kap. 7: Zufallsweg auch als Räuber-Beute-Simulation (Ameisensimulation)</p> <p><i>Schriek:</i> Kap. 8: Vektorgrafik-Projekt</p>

<p>3. Ist-Beziehung</p> <ul style="list-style-type: none"> • Spezialisierung durch zusätzliche Dienste • Generalisierung und Polymorphie • späte Bindung <p>4. Beziehungsdiagramme Diagramme als Werkzeug bei der</p> <ul style="list-style-type: none"> • Modellierung und • Implementierung <p>komplexerer Programme mit mehreren Klassen</p> <p>5. Sichtbarkeitsbereiche</p> <ul style="list-style-type: none"> • Geheimnisprinzip • <i>private</i>-Deklaration insbesondere für Attribute • <i>public</i>-Deklaration als Regelfall für Methoden • <i>get</i>- und <i>set</i>-Methoden 	<p>Eigenschaften, ihre Operationen und ihre Beziehungen (M),</p> <ul style="list-style-type: none"> • 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), • 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 	<p>mit abstrakter Oberklasse</p> <p>z.B. auch <i>Schriek</i>: Kap.6: Figurstift</p>
--	---	---

	<p>grafisch dar (D),</p> <ul style="list-style-type: none">• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).	
--	--	--

Unterrichtsvorhaben EF-V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

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

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit einem ersten Einblick in Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in Java. Zur Untersuchung von Algorithmen bietet sich hier eher eine interpretierte Sprache wie Python an (Pythonprogramm als „ausführbarer Pseudocode“).

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

Daran anschließend werden die erarbeiteten Strategien systematisiert und in Pseudocode (und/oder Python) notiert. Die Schülerinnen und Schüler sollen auf diese Weise mindestens eine Form des *Sortierens durch Vertauschen* und des *Sortierens durch Auswählen* kennen lernen.

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

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Inhalte	Zu entwickelnde Kompetenzen	Beispielkontexte, Medien, Materialien
1. Explorative Erarbeitung eines Sortierverfahrens (a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus	Die Schülerinnen und Schüler <ul style="list-style-type: none">beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),	<i>Beispiel:</i> Sortieren mit Waage Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können.

<p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p> <p>2. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablageplatzes oder mehrerer neuer Ablageplätze)</p> <p>(e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>3. Binäre Suche auf sortierten Daten</p> <p>(a) Suchaufgaben im Alltag und im Kontext informatischer Systeme</p> <p>(b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche</p> <p>(c) Effizienzbetrachtungen zur binären Suche</p>	<p>Speicherplatzbedarf (A),</p> <ul style="list-style-type: none"> entwerfen einen weiteren Algorithmus zum Sortieren (M), analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D). 	<p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: www.csunplugged.org/sorting-algorithms abgerufen: 30. 03. 2014</p> <p>Testläufe von Algorithmen können im Python-Übersetzer leicht durchgeführt, beobachtet und auch zeitgestoppt werden, um ein Gefühl für die Effizienz zu bekommen.</p> <p><i>Materialien:</i> Computer science unplugged – Sorting Algorithms, URL: www.csunplugged.org/sorting-algorithms abgerufen: 30. 03. 2014</p> <p><i>Beispiel:</i> Simulationsspiel zur binären Suche nach Tischtennisbällen Mehrere Tischtennisbälle sind nummeriert, sortiert und unter Bechern verdeckt. Mit Hilfe der binären Suche kann sehr schnell ein bestimmter Tischtennisball gefunden werden.</p> <p><i>Materialien:</i> Computer science unplugged – Searching Algorithms, URL: www.csunplugged.org/searching-algorithms, abgerufen: 30. 03. 2014</p>
--	--	---

Unterrichtsvorhaben EF-VI

Thema: *Geschichte der digitalen Datenverarbeitung und aktuelle Aspekte 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 oder zu aktuellen 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 soll gemeinsam unter dem Aspekt des Datenschutzes und der gesellschaftlichen Auswirkungen über ausgewählte aktuelle Entwicklungen diskutiert werden. Ausgangspunkt können vorangegangene Präsentationsthemen sein, oder auch aktuelle Entwicklungen, die zeigen, wie technische Entwicklungen für das Leben des Einzelnen in der Gesellschaft sowohl positiv bereichernde als auch einschränkende Auswirkungen mit sich bringen können.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens:

Inhalte	Zu entwickelnde Kompetenzen	Beispielkontexte, Medien, Materialien
<p>1. Geschichte der Datenverarbeitung</p> <ul style="list-style-type: none">• Vorläufer und Vordenker (Leibniz, Babbage,...)• Die Anfänge der Digitaltechnik ab dem 2. Weltkrieg• Entwicklungen der Hardware nach der Einführung des Transistors• Entwicklungen im Bereich der Software	<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),	<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>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.</p>

<p>2. Auswirkungen der Automatisierung</p> <ul style="list-style-type: none"> • Änderungen in der Wirtschaft der Industrieländer • Globalisierung und Kommunikationstechnik <p>3. Datenschutz und persönliche Freiheit</p> <ul style="list-style-type: none"> • Überwachungsstaat? • Das Netz vergisst nichts. • Die Macht der Information • Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler • Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“ 	<ul style="list-style-type: none"> • • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K). 	<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 (Download EF-VI.1)</p>
---	--	---