Schulinternes Curriculum

Informatik

Einführungsphase

Cornelius-Burgh-Gymnasium Erkelenz

Stand: 12.05.2014

Ziele und Aufgaben des Faches sind in den Richtlinien genau beschrieben:

http://www.standardsicherung.schulministerium.nrw.de/lehrplaene/upload/klp_SII/if/GOSt_Informatik_Endfassung.pdf

Kompetenzen und Inhaltsfelder

Die Kompetenzen und Inhaltsfelder aus den Richtlinien sind hier angegeben und entsprechen den Beschreibungen des Kernlehrplans, ebenso die Kompetenzerwartungen.

Kompetenzbereiche und Inhaltsfelder des Faches
Im Informatikunterricht erwerben die Lernenden eine Vielzahl von fachbezogenen Fähigkeiten und Fertigkeiten, die sich **fünf Kompetenzbereichen** zuordnen lassen. Diese sind nicht trennscharf, sondern haben vielfältige Bezüge zueinander. Im Sinne eines umfassenden Kompetenzerwerbs ist es daher nicht möglich, den Unterricht nach den einzelnen Kompetenzbereichen sequentiell zu strukturieren. Angemessen ist eine ganzheitliche Sicht als Grundlage für den Kompetenzerwerb.

Kompetenzbereiche

Die fachbezogenen Kompetenzen, die in Gesamtheit informatische Problemlösekompetenz ausmachen, lassen sich den fünf ***Kompetenzbereichen***, *Argumentieren*, *Modellieren*, *Implementieren, Darstellen und Interpretieren* sowie *Kommunizieren und Kooperieren* zuordnen.

Argumentieren
Informatische Zusammenhänge, Vorgehensweisen, Lösungsansätze und Entwurfsentscheidungen bedürfen der Erläuterung und Begründung, um Transparenz, Nachvollziehbarkeit und Überprüfbarkeit im Diskurs zu gewährleisten. Argumentieren umfasst das Erläutern, Begründen und Beurteilen in informatischen Sachzusammenhängen und Prozessen.
bedeutet, einen Sachverhalt zu veranschaulichen und verständlich zu machen. Unter Begründen wird die Darlegung von rational nachvollziehbaren Argumenten auf der Grundlage von Begriffen, Regeln, Methoden und Verfahren der Informatik verstanden. Dazu gehört auch, den Begründungszusammenhang durch geeignete Beispiele zu veranschaulichen. Beurteilen meint, zu einem informatischen Sachverhalt oder Prozess ein selbstständiges Urteil unter Verwendung von Fachwissen und Fachmethoden zu formulieren und zu begründen. Argumentieren umfasst auch die Bewertung von Nutzen, Grenzen und Auswirkungen von Informatiksystemen.

Modellieren
Um ein Problem aus einem inner- oder außerinformatischen Kontext lösen zu können, wird in der Regel zunächst ein informatisches Modell entwickelt, das auf einem prozessorgesteuerten Gerät implementiert werden kann. Informatisches Modellieren zielt auf eine abstrahierende Beschreibung der wesentlichen Komponenten und Parameter eines realen oder geplanten Systems sowie des Ordnungsgefüges und der Wirkungsbeziehungen zwischen ihnen. Der Modellierungsprozess beginnt mit der Analyse und einer strukturierten Zerlegung des Ausgangsproblems. Teilkomponenten müssen identifiziert, konstruiert und gegebenenfalls miteinander vernetzt werden. Ein Ergebnis eines Modellierungsprozesses ist in der Regel eine formale, textuelle oder grafische Darstellung.

Implementieren
Implementieren umfasst die Umsetzung eines Modells in ein Informatiksystem. Dazu gehören das Programmieren, Evaluieren und Validieren von Modellbestandteilen unter Nutzung geeigneter Werkzeuge. Grundlegende Methoden und Denkweisen der Programmentwicklung werden dabei in den Vordergrund gestellt. Die Programmerstellung ist ein bedeutsamer Bestandteil des Problemlösungsprozesses, weil erst dadurch das Modell wirksam wird. An dem entstandenen Informatiksystem können Wirkungen der Modellentscheidungen diskutiert sowie Ursachen und Tragweite von möglichen Fehlern im Modell erkannt und korrigiert werden. Dadurch werden die Selbstreflexion des Lösungsprozesses und eine vertiefte Modellkritik unterstützt.

Darstellen und Interpretieren
Die Informatik hat zur Unterstützung von Problemlöse- und Modellbildungsprozessen ein reiches Repertoire an Darstellungsformen entwickelt. Schülerinnen und Schüler werden nach und nach mit unterschiedlichen Darstellungsformen konfrontiert, die sie in inner- und außerinformatischen Kontexten selbst nutzen. Vorgegebene Darstellungen müssen anwendungsbezogen interpretiert werden. Im Rahmen eigener Problemlösungen müssen angemessene Darstellungsformen unter Verwendung der fachspezifischen Notation angewendet werden. Dies fördert ein Verständnis von Zusammenhängen und Bezügen zwischen unterschiedlichen informatischen Sachverhalten sowie die Fähigkeit, diese anderen deutlich zu machen.

Kommunizieren und Kooperieren
Die Kenntnis und Nutzung arbeitsteiliger und kooperativer Vorgehensweisen ist für die Entwicklung komplexer Informatiksysteme erforderlich, um prozessorientiertes Arbeiten zu planen und abzusichern. Zum Kommunizieren im Sinne eines fachlichen Austausches gehören die sachadäquate Darstellung und Dokumentation zur Weitergabe von Sachverhalten sowie die Nutzung geeigneter Werkzeuge, die die Kommunikation unterstützen. Für eine sachangemessene und präzise Verständigung über informatische Gegenstände sind ein angemessener Umgang mit Fachbegriffen und der sukzessive Aufbau einer Fachsprache unerlässlich.

Inhaltsfelder
Kompetenzerwerb ist an fachliche Inhalte gebunden. Die für den Informatikunterricht obligatorischen Inhalte, an denen die Kompetenzen entwickelt werden sollen, lassen sich den folgenden fünf Inhaltsfeldern zuordnen. Diese werden bereits in der Einführungsphase aufgegriffen und in der Qualifikationsphase vertieft.

Daten und ihre Strukturierung
Die automatische Verarbeitung von Informationen mittels Maschinen ist überhaupt erst durch deren digitale Repräsentation in Form von Daten möglich. Für die rechnergestützte Lösung von Problemen in inner- und außerinformatischen Kontexten müssen daher Informationen in angemessener Struktur durch Daten und zugehörige Operationen repräsentiert werden, so dass die Daten zielgerichtet und effizient automatisch verarbeitet und die Ergebnisse wiederum als Information interpretiert werden können.

Algorithmen
Zu vielen bedeutenden wissenschaftlichen Erfolgen und technischen Errungenschaften der jüngeren Zeit hat die Informatik maßgeblich beigetragen. Neben der rasanten Steigerung der Leistungsfähigkeit der technischen Systeme sind diese Fortschritte insbesondere der Entwicklung

von innovativen und effizienten Algorithmen zu verdanken. Ein Algorithmus ist eine genaue Beschreibung von Handlungsschritten zur Lösung eines Problems, die von einem Prozessor ausgeführt werden können. Häufig verwendete Grundkonstrukte von Algorithmen sowie Algorithmen, die im Kontext bestimmter Problemklassen von elementarer Bedeutung sind, lassen sich unter Berücksichtigung ihrer Effizienz adaptieren, um neue Aufgabenstellungen in konkreten Anwendungskontexten problemgerecht einer automatischen Verarbeitung zuzuführen.

Formale Sprachen und Automaten
Der Einsatz von Informatiksystemen zur Lösung komplexer Probleme ist nur unter Verwendung formaler Sprachen als Mittler zwischen Mensch und Maschine möglich. Sprachen dienen zur Kommunikation und genügen Regeln zur Bildung von Wörtern und Sätzen. Formale Sprachen der Informatik werden durch Grammatiken präzise beschrieben. Zu formalen Sprachen können Automaten entwickelt werden, die die Wörter der Sprache akzeptieren oder weiterverarbeiten. Eine fachliche Beschreibung von Automaten mithilfe einer Menge von Zuständen samt Regeln für die zeitliche Abfolge von Zustandsübergängen ist als Modellierungstechnik in verschiedenen Problemfeldern anwendbar. Automaten eigenen sich in besonderem Maße, um mit Hilfe theoretischer Betrachtungen auch die Grenzen von Automatenmodellen zu beleuchten.

Informatiksysteme
Informatiksysteme sind heute weltweit miteinander vernetzt. Ein Informatiksystem ist eine spezifische Zusammenstellung von Hardware-, Software- und Netzwerkkomponenten zur Lösung eines Anwenderproblems. Gegenstand der Betrachtung in diesem Inhaltsfeld sind schwerpunktmäßig der prinzipielle Aufbau singulärer und vernetzter Rechnersysteme und deren Interaktion untereinander und mit dem Benutzer.

Informatik, Mensch und Gesellschaft
Informatiksysteme stehen in intensiver Wechselwirkung mit Individuum und Gesellschaft. Ihr Einsatz hat weitreichende Konsequenzen für unsere Lebens- und Arbeitswelt. Handlungsspielräume müssen im Spannungsfeld von Rechten und Interessen des Individuums, gesellschaftlicher Verantwortung und möglichen Sicherheitsrisiken wahrgenommen werden.

Unterrichtsvorhaben EF-I (Zeitbedarf ca. 6 Stunden)

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten und Teilgebiete der Fachwissenschaft Informatik

Leitfragen: *Womit beschäftigt sich die Wissenschaft der Informatik? Wie gliedert sich die Fachwissenschaft? Welche fundamentalen Konzepte müssen Fachwissenschaftler in ihre Arbeit einbeziehen, damit informatische Systeme funktionieren? Wie kann man die in der Schule vorhandene Ausstattung genutzt werden?*

Vorhabenbezogene Konkretisierung: Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik und seine fünf Inhaltsfelder 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.

Das Unterrichtsvorhaben sollte so strukturiert werden, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) sich größtenteils selbstständig erarbeiten und nachvollziehen können. Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und Gesellschaft in groben Zügen überblicksartig thematisiert. Das so erworbene Wissen kann auf weitere, den Schülerinnen und Schülern bekannte Informatiksysteme, übertragen werden.

In einem weiteren Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen bereits so betrachtet werden, dass ein kompetenter Umgang mit diesem ermöglicht wird. Auch bietet sich hier die Thematisierung der Materialverzeichnisnutzung auf der Schulhomepage an.

Sequenzierung	Zu entwickelnde Kompetenzen
1. Informatiksysteme und ihr genereller Aufbau a) Daten und ihre Strukturierung b) Algorithmen c) Formale Sprache und Automaten d) Informatiksysteme e) Informatik, Mensch und Gesellschaft 2. Der kompetente Umgang mit dem Schulnetzwerk / Intranet a) Erstellen und Anlegen von Ordnerstrukturen b) Sortieren von Dateien und Ordnern c) Eingabe von Befehlen über die Eingabeaufforderung bzw. Shell d) Einzelrechner und Netzwerke, Zugangsdaten e) Sicherheit und Datenschutz	Die Schülerinnen und Schülerinnen • werden sich der Auswirkungen der Nutzung von Informatiksystemen bewusst (A). • nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K). • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst. • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).

Unterrichtsvorhaben EF-II (Zeitbedarf ca. 8 Stunden)

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen: *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 Objektdiagramme und Klassendiagramme, eingeführt.

Im Anschluss wird die objektorientierte Analyse für ein Greenfoot-Szenario durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in diesem Stadium zudem 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 oder zu Hohe Herausforderungen darstellen. Natürlich kann die Arbeit an diesen Projekten unmittelbar in ein folgendes Unterrichtsvorhaben (u. a. Kontrollstrukturen) münden.

Sequenzierung	Zu entwickelnde Kompetenzen
1. Identifikation von Objekten und Klassen a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. b) Objekte werden durch Objektdiagramme (z. B. UML), Klassen durch Klassendiagramme dargestellt. 2. Analyse von Objekten und Klassen im Greenfoot-Szenario a) Schritte der objektorientierten Analyse, Modellierung und Implementation b) Analyse und Erprobung der Objekte in Greenfoot-Szenario 3. Implementation einfacher Aktionen in Greenfoot	Die Schülerinnen und Schülerinnen • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M). • modellieren Klassen mit ihre Attributen, ihren Methoden und Assoziationsbeziehungen (M). • modellieren Klassen mit ihre Attributen und Methoden, ggf. bereits unter Verwendung von Vererbung (M). • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M).

a) Quelltext einer Java-Klasse b) Implementation eigener Methoden, Dokumentation mit Java-Doc c) Programme übersetzen (Aufgabe des Compilers) und testen / Fehlerkorrektur	• stellen den Zustand eines Objekts dar (D). • stellen die Kommunikation zwischen Objekten graphisch dar (M). • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I). • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

Unterrichtsvorhaben EF-III (Zeitbedarf ca. 18 Stunden)

Thema: Erweiterung der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Leitfragen: *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

Vorhabenbezogene Konkretisierung:

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen der Wiederholung und der bedingten Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden auch systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des vorliegenden Unterrichtsvorhabens liegt auf der Nutzung von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse (unter Berücksichtigung des Geheimsnisprinzips) realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden und werden für eventuelle Fallstricke sensibilisiert.

Sequenzierung	**Zu entwickelnde Kompetenzen**
1. Algorithmen a) Wiederholungen (While-Schleife / for-Schleife) b) bedingte Anweisungen / Verzweigungen c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme 2. Variablen und Methoden a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife b) Implementierung eigener Methoden mit Parameterübergabe und / oder Rückgabewert c) Implementierung von Konstruktoren d) Realisierung von Attributen	Die Schülerinnen und Schülerinnen • 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, Parameter 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 Objektes dar (D). • stellen die Kommunikation zwischen Objekten graphisch dar (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen graphisch 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. • analysieren und erläutern einfache Algorithmen und Programme (A). • modifizieren einfache Algorithmen und Programme (I). • entwerfen einfache Algorithmen und stellen sie umgangsprachlich und graphisch dar (M). • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I). • testen Programme schrittweise anhand von Beispielen (I). • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache oder auch in Pseudocode (I). • interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

Unterrichtsvorhaben EF-IV (Zeitbedarf ca. 18 Stunden)

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen

Leitfrage: *Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert? Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?*

Vorhabenbezogene Konkretisierung: Das Unterrichtsvorhaben hat die Entwicklung von Objekt- und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler (nochmals) Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung z. B. eines Spiels kann ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang kann auch das Erstellen von graphischen Benutzeroberflächen eingeführt werden. Zum Abschluss kann kurz auf das Prinzip der abstrakten Klasse eingegangen werden. Dies ist aber für die Einführungsphase nicht obligatorisch.

Sequenzierung	Zu entwickelnde Kompetenzen
1. Umsetzung von Anforderungen in Entwurfsdigramme a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert 2. Implementationsdiagramme als erster Schritt der Programmierung a) Erweiterung des Entwurfsdiagramms um Konstrukturen und get- und set-Methoden b) Festlegung von Datentypen in Java, sowie Rückgaben und Parametern	Die Schülerinnen und Schülerinnen • 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 einfach Datentypen, Objekttypen oder lineare Datensammlungen zu (M). • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M).

c) Entwicklung von Klassendokumentationen d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung 3. Programmierung anhand der Dokumenation und des Implementations- und Sequenzdiagrammes a) Klassen werden in Java-Quellcode umgesetzt b) Das Geheimnisprinzip wird umgesetzt c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumenationen auf ihre Korrektheit überprüft 4. Vererbungsbeziehungen a) Das Grundprinzip der Vererbung wird erarbeitet b) Die Vorteile der Vererbungsbeziehungen c) Vererbung wird implementiert 5. Softwareprojekt a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) b) Erweiterung des Spiels um weitere Funktionalitäten c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer graphischen Benutzeroberfläche d) ggf. (arbeitsteilige) Implementation des / eines Spiels	• stellen den Zustand eines Objekts dar (D). • stellen die Kommunikation zwischen Objekten graphisch dar (M). • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen graphisch 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). • implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I). • testen Programme schrittweise anhand von Beispielen (I).

Unterrichtsvorhaben EF-V (Zeitbedarf ca. 15 Stunden)

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfragen: *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

Vorhabenbezogene Konkretisierung: Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, so dass eine entsprechende Implementierung nicht zwingend erforderlich ist.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. In den Einstiegen sollten die jeweiligen Strategien handlungsorientiert erkundet und intuitive Effizienzbetrachtungen der Suchalgorithmen vorgenommen werden.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Sequenzierung	Zu entwickelnde Kompetenzen
1. Modellierung und Implementation von Datenansammlungen a) Modellierung von Attributen als Felder b) Deklaration, Instanziierung und Zugriffe auf ein Feld 2. Explorative Erarbeitung von Suchverfahren a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion b) Vergleich der Verfahren durch intuitive Effizienzbetrachtungen 3. Systematisierung von Algorithmen und Effizienzbetrachtungen a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche	Die Schülerinnen und Schülerinnen • analysieren Such- und Sortieralgorithmen und werden sie an Beispielen an (D). • entwerfen einen weiteren Algorithmus zum Sortieren (M). • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A). • modifizieren einfache Algorithmen und Programme (I). • entwerfen einfache Algorithmen und stellen sie umgangssprachlich und graphisch dar (M).

d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs e) Analyse eines weiteren Sortieralgorithmus, sofern nicht bereits geschehen	

Unterrichtsvorhaben EF-VI (Zeitbedarf ca. 8 Stunden)

Thema: Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Kodierung und den strukturellen Aufbau eines Informatiksystems

Leitfragen: *Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?*

Vorhabenbezogene Konkretisierungsmöglichkeiten: Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt. Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Anschließend wird auf den Begriff der Information eingegangen und die Möglichkeit der Kodierung in Form von Daten thematisiert. Ferner 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 soll jeweils ein Bezug zur konkreten Nutzung der informatischen Ausstattung der Schule hergestellt werden. So wird in die verantwortungsvolle Nutzung dieser Systeme vertieft.

Sequenzierung	**Zu entwickelnde Kompetenzen**
1. Analoge und digitale Aufbereitung und Verarbeitung von Daten a) Erarbeitung der Unterschiede von analog(er) und digital(er) Darstellung b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital c) Informatik als Wissenschaft der Verarbeitung von Informationen 2. Der Umgang mit binärer Codierung von Informationen a) Das binäre (und hexadezimale) Zahlensystem b) binäre Informationsspeicherung c) binäre Verschlüsselung (ggf.)	Die Schülerinnen und Schülerinnen • 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). • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A). • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).

d) Implementation eines Binärumrechners (ggf.) 3. Aufbau informatischer Systeme / Netzstrukturen a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systeme b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips c) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation	• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I).

Unterrichtsvorhaben EF-VII (Zeitbedarf ca. 10 Stunden)

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

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

Vorhabenbezogene Konkretisierungsmöglichkeiten:

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt. Hier können Aspekte der Kryptologie einfliesen. In einem ersten Schritt des Unterrichtsvorhabens wird daher folglich anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können. Dieses Unterrichtsvorhaben kann sehr stark auf Schülerrecherchen mit dem Ziel von Schülervorträgen bzw. schriftlichen Ausarbeitungen aufbauen.

Sequenzierung	Zu entwickelnde Kompetenzen
1. Schriftzeichen, Rechenmaschinen und Computer a) Anhand von Schwerpunkte, wie z. B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden 2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen	Die Schülerinnen und Schülerinnen • nutzen das Internet zur Recherche, zum Datentausch und zur Kommunikation (K). • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A). • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).

Nachteile betrachtet 3. Vertiefung des Themas Datenschutz a) Erarbeitung grundlegender Begriffe des Datenschutzes b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“ (Materialblatt)	