

Schulinterner Lehrplan für das Fach Informatik am Quirinus-Gymnasium

Stand: 01.06.2014

Schulisch-organisatorische Voraussetzungen:

Das Quirinus-Gymnasium Neuss ist ein städtisches Gymnasium mit ca. 1100 Schülerinnen und Schülern sowie ca. 100 Kolleginnen und Kollegen. Das Einzugsgebiet besteht im Wesentlichen aus der Neusser Innenstadt, allerdings kommen auch viele Schülerinnen und Schüler aus dem Neusser Umland. Die Schule hat einen altsprachlichen und musisch-künstlerischen Schwerpunkt, den Naturwissenschaften oder der informationstechnischen Ausbildung wird traditionell eher eine untergeordnete Rolle beigemessen.

Der Fachschaft Informatik gehören zurzeit drei Kollegen an.

Das Fach Informatik wird ab der 8. Klasse im Rahmen der differenzierten Mittelstufe dreistündig angeboten, hier in Kombination mit dem Fach Mathematik, sowie in Grundkursen im Bereich der gymnasialen Oberstufe. Leistungskurse können aufgrund der personellen Situation nicht angeboten werden.

Die Kursstärken schwanken im Mittelstufenbereich zwischen 15 und 25 Schülerinnen und Schülern, insgesamt kann ein leichter Rückgang der Belegungszahlen in der Mittelstufe innerhalb der letzten 10 Jahre registriert werden.

Die Fachschaft kann auf zwei Informatikräume zurückgreifen, die mit jeweils 2 Lehrer- und 14 Schülerrechnern sowie Beamern, Präsentationsflächen, einem netzwerkfähigen Drucker sowie einem Novell- und einem Linuxs-Server ausreichend ausgestattet sind.

Der Unterricht erfolgt zur Zeit (Stand Juni 2014) im 45-Minuten-Takt, die Kursblockung für die differenzierte Mittelstufe sowie die Oberstufe sieht grundsätzlich eine Doppelstunde sowie eine Einzelstunde vor.

Seitenaspekte:

Fachübergreifende Projekte werden punktuell (Musik, Mathematik, Kunst) und auf Einzelinitiative von engagierten Kollegen durchgeführt, sind aber zurzeit nicht konzeptionell verankert.

Außerhalb des Fachbereichs Informatik sollen alle Schüler mit Hilfe von Selbstlernmodulen (gekoppelt an die Fachschaften Mathematik, Deutsch und die Naturwissenschaften) an die Bereiche Tabellenkalkulation, Textverarbeitung und Präsentationstechniken herangeführt werden, die Umsetzung und Dokumentation der entsprechenden Maßnahmen erfolgt allerdings nur rudimentär. Ein Virtueller Klassenraum, in dem Lehrerinnen und Lehrer Lerngruppen anlegen können und Dokumente oder Links zu Fachinhalten zur Verfügung stellen können und mit der Lerngruppe in kommunizieren können, existiert zwar, wird allerdings ebenfalls nicht flächendeckend, sondern eher rudimentär genutzt (von ca. 10% der Kollegen).

Informationsheoretische Inhalte:

Das Fach Informatik wird am Quirinus-Gymnasium seit Mitte der 80er Jahre unterrichtet, hat also eine lange Tradition. Ein grundlegendes Paradigma bestand darin, nicht schnell veraltende Inhalte, sondern informationstheoretische Grundlagen zu unterrichten. So werden in der Mittelstufe Inhalte der Aussagenlogik und der technischen Informatik den Schülerinnen und Schülern nahe gebracht sowie eine Einführung in die Programmierung. Da in der differenzierten Mittelstufe nicht alle

Schülerinnen und Schüler das Fach Informatik belegen, muss ab der Einführungsphase darauf Wert gelegt werden, Grundlagen für alle Schülerinnen und Schüler zu legen, um in der Qualifikationsphase erfolgreich mitzuarbeiten. Bislang wurde in der Einführungsphase die Programmiersprache Pascal unterrichtet, um prozedurale Methoden und grundlegende Merkmale von Programmiersprachen (Datentypen, Entscheidungsstrukturen, Wiederholungsanweisungen, Listen und Felder, Funktionen, Prozeduren, Parameter, Sortiermethoden) einzuführen, die die Grundlagen aller Programmiersprachen bilden und ein algorithmisches Grundverständnis vermitteln sollten. In der Qualifikationsphase wurde dieses Konzept bislang um dynamische Datenstrukturen wie Pointer, Listen und Binärbäume erweitert und mit Methoden der Assemblerprogrammierung (Rekursion) vertieft. Eine Einführung in Datenbankmodelle sowie Aspekte der theoretischen Informatik wurden bislang fakultativ eingeführt.

Um das Konzept der prozeduralen Programmierung zu erweitern und ein tieferes Verständnis für moderne Programmierparadigmen zu legen, wurde im letzten Jahr die Objektorientierte Programmierung mit Delphi eingeführt.

Die neuen Kernlehrpläne sowie die Einschränkung der Programmierumgebung auf die Programmiersprache JAVA macht eine konzeptionelle Änderung notwendig, auf die dieser schulinterne Lehrplan reagieren soll.

Problematisch erscheint hier im Wesentlichen der Einstieg, da JAVA ohne das objektorientierte Konzept nicht auskommt und Schülerinnen und Schülern ohne Vorerfahrung ein verständlicher Weg bereitet werden muss, um auch die Ebene der Implementation zu erreichen.

Übersichtsraster Unterrichtsvorhaben in der Einführungsphase Teil 1

Einführungsphase	
<p><u>Unterrichtsvorhaben E-I</u></p> <p>Thema:</p> <p><i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Einsatz, Nutzung und Aufbau von Informatiksystemen - Wirkung der Automatisierung <p>Zeitbedarf: 6 Stunden</p>	<p><u>Unterrichtsvorhaben E-II</u></p> <p>Thema:</p> <p><i>Grundlagen der Programmierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung, Variablenkonzept - Eingabe-, Ausgabeoperationen, Bedingungen, verschachtelte Bedingungen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Syntax und Semantik einer Programmiersprache - lokale, globale Variablen, Setter- und getter-Methoden, Parameter, Gültigkeit von Variablen <p>Zeitbedarf: 12 Stunden</p>
<p><u>Unterrichtsvorhaben E-III</u></p> <p>Thema:</p> <p><i>Kontrollstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <p>Widerholungsanweisungen</p> <p>Inhaltliche Schwerpunkte:</p> <p>For-, while, repeat-Schleifen</p> <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben E-IV</u></p> <p>Thema:</p> <p>Listen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Implementieren - Darstellen und Interpretieren - Argumentieren <p>Inhaltsfelder:</p> <p>Ein- und zweidimensionale Arrays</p> <p>Inhaltliche Schwerpunkte:</p> <p style="padding-left: 40px;">Ein- und Ausgabe von Listen Sortieren, zentrale Operationen (Suche, Maximum, Minimum)</p> <p>Zeitbedarf: 8 Stunden</p>

Einführungsphase	
<p><u>Unterrichtsvorhaben E-V</u></p> <p>Thema:</p> <p><i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 8 Stunden</p>	<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema:</p> <p><i>Algorithmische Grundstrukturen in Java</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>

Übersichtsraster Unterrichtsvorhaben in der Einführungsphase Teil 2

Einführungsphase	
<p><u>Unterrichtsvorhaben E-VI</u></p> <p>Thema:</p> <p><i>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Kommunizieren und Kooperieren - Darstellen und Interpretieren 	<p><u>Unterrichtsvorhaben E-VII</u></p> <p>Thema:</p> <p><i>Graphikkonzepte in BlueJ</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren

<ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 18 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - <p>Inhaltliche Schwerpunkte:</p> <p>Zeitbedarf: 9 Stunden</p>
	<p>Summe Einführungsphase:</p> <p>77 Stunden</p>

Unterrichtsvorhaben EF-I

Thema: Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik

Leitfragen: Was macht Informatik? Welche fundamentalen Konzepte müssen Informatikerinnen und Informatiker in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk- und Computersystem wiederfinden?

Vorhabenbezogene Konkretisierung:

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes 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 thematisiert. Das am Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Informatiksysteme und ihr genereller Aufbau (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)	Kapitel 1 „Was macht Informatik“ Als Anschauungsmaterial bieten sich Navigationsgeräte an
2. Der kompetente Umgang mit dem Schulnetzwerk (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		Kapitel 1 „Was macht Informatik“ Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

Leitfragen: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario 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 in das Programm BlueJ eingeführt. Die Lernenden implementieren und testen einfache Programme. Die BlueJ-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.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Identifikation von Objekten und Klassen</p> <p>(a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</p> <p>(c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I). 	<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.1 Objektorientierte Modellierung</p>
<p>2. Analyse von Objekten und Klassen mit BlueJ</p> <p>(a) Schritte der objektorientierten Analyse, Modellierung und Implementation</p> <p>(b) Analyse und Erprobung der Objekte</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.2 Objekte in BlueJ</p> <p>Von der Realität zu Objekten</p> <p>Von den Objekten zu Klassen, Klassendokumentation</p> <p>Objekte inspizieren</p> <p>Methoden aufrufen</p> <p>Objektidentität und Objektzustand</p>
<p>3. Implementierung einfacher Programme mit BlueJ</p> <p>(a) Quelltext einer Java-Klasse</p> <p>(b) Implementation eigener Methoden, Dokumentation mit JavaDoc</p> <p>(c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p>Kapitel 2 Einführung in die Objektorientierung</p> <p>2.3 Programmierung in BlueJ</p> <p>Methoden schreiben</p> <p>Programme übersetzen und testen</p>

Unterrichtsvorhaben EF-III

Thema: Algorithmische 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 Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz 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 realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), - modifizieren einfache Algorithmen und Programme (I),	<i>Kapitel 3 Algorithmen</i> 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 Algorithmen entwickeln

<p>2. Variablen und Methoden</p> <p>(a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife</p> <p>(b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert</p> <p>(c) Implementierung von Konstruktoren</p> <p>(d) Realisierung von Attributen</p>	<ul style="list-style-type: none"> - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (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>Kapitel 4 Variablen und Methoden</p> <p>4.1 lokale Variablen</p> <p>4.2 Methoden</p> <p>4.3 Attribute</p>
---	--	---

Unterrichtsvorhaben EF-IV

Thema: Das ist die digitale Welt! - Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

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 Konkretisierung:

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 als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der von-Neumann-Architektur erarbeitet werden.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Analoge und digitale Aufbereitung und Verarbeitung von Daten (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - stellen ganze Zahlen und Zeichen in Binärcodes dar (D), - interpretieren Binärcodes als Zahlen und Zeichen (D)	<i>Exkurs „Analog und Digital“</i>

2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners	- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	<i>Exkurs „Binäre Welt“</i>
3. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips		<i>Exkurs „Arbeitsweise eines Computers“</i>

Unterrichtsvorhaben EF-V

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

Leitfragen: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

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 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 eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz

kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren und erläutern eine objektorientierte Modellierung (A), - stellen die Kommunikation zwischen Objekten grafisch dar (M), - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), - modellieren Klassen unter Verwendung von Vererbung (M), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), 	<p>Kapitel 6 Klassenentwurf 6.1. Von der Realität zum Programm 6.2. Objekte identifizieren 6.3. Klassen und Beziehungen entwerfen</p>
<p>2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden (b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Erstellung von Sequenzdiagrammen als Vorbereitung Vorbereitung für die Programmierung</p>	<ul style="list-style-type: none"> - 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). - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) 	<p>Kapitel 6 Klassenentwurf 6.4 Klassen und Beziehungen implementieren 6.5 Vererbung</p>
<p>3. Programmierung anhand der Dokumentation 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 Dokumentationen auf ihre Korrektheit überprüft.</p>	<ul style="list-style-type: none"> - 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). - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D) 	<p>Kapitel 6 Klassenentwurf 6.4 Klassen und Beziehungen implementieren 6.5 Vererbung</p>
<p>4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert</p>		<p>Kapitel 6 Klassenentwurf 6.5 Vererbung</p>

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 grafischen Benutzeroberfläche
- (d) (arbeitsteilige) Implementation des Spiels

Kapitel 8 Softwareprojekte

- 8.1 Softwareentwicklung
- 8.2 Oberflächen

Unterrichtsvorhaben EF-VI

Thema: Such- und Sortialgorithmen

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

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortialgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

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. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens: nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p>1. Modellierung und Implementation von Datenansammlungen</p> <p>(a) Modellierung von Attributen als Felder</p> <p>(b) Deklaration, Instanziierung und Zugriffe auf ein Feld</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 Zeit und Speicherplatzbedarf (A) 	<p><i>Kapitel 7 Sortieren und Suchen auf Feldern</i></p> <p>7.1 Das Feld – Eine Sammlung von Daten</p>
<p>2. Explorative Erarbeitung von Suchverfahren</p> <p>(a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion.</p> <p>(b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.</p>	<ul style="list-style-type: none"> - ordnen Attributen lineare Datenansammlungen zu (M) 	<p><i>Kapitel 7 Sortieren und Suchen auf Feldern</i></p> <p>Projekteinstieg 1: Suchen</p> <p>7.2 Suchen mit System</p> <p>Lineare Suche</p> <p>Binäre Suche</p> <p>Hashing</p>
<p>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p> <p>(e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p><i>Kapitel 7 Sortieren und Suchen auf Feldern</i></p> <p>Projekteinstieg 2: Sortieren</p> <p>7.3 Ordnung ist das halbe Leben!? – Sortieren</p> <p>Sortieren</p> <p>Selection Sort</p> <p>Insertion Sort</p> <p>Bubble Sort</p>

Unterrichtsvorhaben EF-VII

Thema: Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Leitfragen: Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?

Vorhabenbezogene Konkretisierung:

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.

Im ersten Schritt des Unterrichtsvorhabens werden 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.

Zeitbedarf: 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Schriftzeichen, Rechenmaschine, Computer (a) Anhand von Schwerpunkten, 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.	Die Schülerinnen und Schüler <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)• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)	<i>Exkurs „Geschichte der Informatik“</i>

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 Nachteile betrachtet.

Exkurs „Informatik und Gesellschaft“