

Schulinterner Lehrplan

zum Kernlehrplan für die gymnasiale Oberstufe

der Heinrich-Böll-Gesamtschule

Informatik

(Stand: 18.09.2015)

Inhaltsverzeichnis

Übersichtsraster Unterrichtsvorhaben.....	3
Einführungsphase.....	3
Qualifikationsphase 1.....	5
Qualifikationsphase 2.....	6
Konkretisierte Unterrichtsvorhaben.....	8
Einführungsphase.....	8
Unterrichtsvorhaben EF-I.....	8
Nutzung von Informatiksystemen.....	8
Unterrichtsvorhaben EF-II.....	9
Grundlagen der objektorientierten Analyse.....	9
Unterrichtsvorhaben EF-III.....	10
Objektorientierte Analyse.....	10
Unterrichtsvorhaben EF-IV.....	11
Such- und Sortieralgorithmen.....	11
Unterrichtsvorhaben EF-V.....	12
Die digitale Welt.....	12
Qualifikationsphase.....	13
Unterrichtsvorhaben Q1-I.....	13
Wiederholung der objektorientierten Modellierung.....	13
Unterrichtsvorhaben Q1-II.....	14
Rekursive Programmierung.....	14
Unterrichtsvorhaben Q1-III.....	14
Anwendungen mit dynamischen, linearen Datenstrukturen.....	14
Unterrichtsvorhaben Q1-IV.....	15
Suchen und Sortieren auf linearen Datenstrukturen.....	15
Unterrichtsvorhaben Q1-V.....	16
Modellierung und Nutzung von relationalen Datenbanken.....	16
Unterrichtsvorhaben Q1-VI.....	17
Sicherheit und Datenschutz in Netzstrukturen.....	17
Unterrichtsvorhaben Q2-I.....	18
Anwendungen mit dynamischen, nichtlinearen Datenstrukturen.....	18
Unterrichtsvorhaben Q2-II.....	19
Endliche Automaten und formale Sprachen.....	19
Unterrichtsvorhaben Q2-III.....	19
Grenzen der Automatisierbarkeit.....	19
Unterrichtsvorhaben Q2-IV.....	20
Wiederholung und Vertiefung ausgewählter Kompetenzen.....	20

Übersichtsraster Unterrichtsvorhaben

I) Einführungsphase

Einführungsphase	
<p><u>Unterrichtsvorhaben EF-I</u></p> <p>Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Informatiksysteme ➤ Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Einzelrechner ➤ Dateisystem ➤ Internet ➤ Einsatz von Informatiksystemen <p>Zeitbedarf: ca. 9 Stunden</p>	<p><u>Unterrichtsvorhaben EF-II</u></p> <p>Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten</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: ca. 15 Stunden</p>
<p><u>Unterrichtsvorhaben EF-III</u></p> <p>Thema: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ 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: ca. 24 Stunden</p>	<p><u>Unterrichtsvorhaben EF-IV</u></p> <p>Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Daten und ihre Strukturierung ➤ Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Algorithmen zum Suchen und Sortieren ➤ Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: ca. 12 Stunden</p>

Einführungsphase

Unterrichtsvorhaben EF-V

Thema:

Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung
- Digitalisierung
- Datenschutz

Zeitbedarf: ca. 15 Stunden

Summe Einführungsphase: 75 Stunden

II) Qualifikationsphase (Q1 und Q2) - GRUNDKURS

Qualifikationsphase 1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Daten und ihre Strukturierung ➤ Algorithmen ➤ Formale Sprachen und Automaten ➤ Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Objekte und Klassen ➤ Analyse, Entwurf und Implementierung von Algorithmen ➤ Syntax und Semantik einer Programmiersprache ➤ Nutzung von Informatiksystemen <p>Zeitbedarf: ca. 9 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: Rekursive Programmierung an mathematischen und grafischen Beispielen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Algorithmen ➤ Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Analyse, Entwurf und Implementierung von Algorithmen ➤ Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: ca. 9 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p>Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ 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 ➤ Analyse, Entwurf und Implementierung von Algorithmen ➤ Algorithmen in ausgewählten informatischen Kontexten ➤ Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: ca. 15 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p>Thema: Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Algorithmen ➤ Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Analyse, Entwurf und Implementierung von Algorithmen ➤ Algorithmen in ausgewählten informatischen Kontexten ➤ Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: ca. 15 Stunden</p>

Qualifikationsphase 1

<p><u>Unterrichtsvorhaben Q1-V</u></p> <p>Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Daten und ihre Strukturierung ➤ Algorithmen ➤ Formale Sprachen und Automaten ➤ Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Datenbanken ➤ Algorithmen in ausgewählten informatischen Kontexten ➤ Syntax und Semantik einer Programmiersprache ➤ Sicherheit <p>Zeitbedarf: ca. 21 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-VI</u></p> <p>Thema: Sicherheit und Datenschutz in Netzwerken</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Informatiksysteme ➤ Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Einzelrechner und Rechnernetzwerke ➤ Sicherheit ➤ Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: ca. 9 Stunden</p>
--	--

Summe Qualifikationsphase 1: 78 Stunden

Qualifikationsphase 2

<p><u>Unterrichtsvorhaben Q2-I</u></p> <p>Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Implementieren ➤ Darstellen und Interpretieren ➤ 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 ➤ Analyse, Entwurf und Implementierung von Algorithmen ➤ Algorithmen in ausgewählten informatischen Kontexten ➤ Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: ca. 24 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u></p> <p>Thema: Endliche Automaten und formale Sprachen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> ➤ Argumentieren ➤ Modellieren ➤ Darstellen und Interpretieren ➤ Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> ➤ Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> ➤ Endliche Automaten ➤ Grammatiken regulärer Sprachen ➤ Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: ca. 20 Stunden</p>
--	--

Qualifikationsphase 2

Unterrichtsvorhaben Q2-III

Thema:

Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen:

- Argumentieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Zeitbedarf: ca. 12 Stunden

Unterrichtsvorhaben Q2-IV

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase

Summe Qualifikationsphase 2: 56 Stunden

Konkretisierte Unterrichtsvorhaben

I) Einführungsphase

Unterrichtsvorhaben EF-I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Information, deren Kodierung und Speicherung</p> <p>(a) Informatik als Wissenschaft der Verarbeitung von Informationen</p> <p>(b) Darstellung von Informationen in Schrift, Bild und Ton</p> <p>(c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner</p> <p>(d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A), ◆ nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), ◆ nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).
<p>2. Informations- und Datenübermittlung in Netzen</p> <p>(a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation</p> <p>(b) Informatische Kommunikation in Rechnernetzen am Beispiel des Schulnetzwerks (z.B. Benutzeranmeldung, Netzwerkordner, Zugriffsrechte, Client-Server)</p> <p>(c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (z.B. Netzwerkadresse, Paketvermittlung, Protokoll)</p> <p>(d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet</p>	
<p>3. 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 und Grundlage der „Von-Neumann-Architektur“</p>	

Unterrichtsvorhaben EF-II

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von einfachen Beispielkontexten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Identifikation von Objekten</p> <p>(a) An einem einfachen Beispiel werden Objekte im Sinne der objektorientierten Modellierung eingeführt.</p> <p>(b) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Klasse zusammengefasst.</p> <p>(c) Klassen werden in Diagrammen visualisiert und mit sinnvollen Attributen und Methoden versehen.</p> <p>(d) Vertiefung: Modellierung weiterer Beispiele ähnlicher Art.</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) ◆ modellieren Klassen mit ihren Attributen und Methoden (M), ◆ stellen Attribute und Methoden in einem Klassendiagramm dar (D), ◆ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu (M), ◆ implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),
<p>2. Implementation einfacher Klassen mit BlueJ</p> <p>(a) Grundaufbau einer Java</p> <p>(b) Deklaration und Initialisierung von Objekten</p> <p>(c) Konstruktoren</p> <p>(d) Einfache Methodenaufrufe</p> <p>(e) Primitive Datentypen als Attribute, Parameter und Variablen</p> <p>(f) Bedingte Anweisungen (if-Abfragen)</p>	<ul style="list-style-type: none"> ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ stellen den Zustand eines Objektes dar (D), ◆ analysieren und erläutern einfache Algorithmen und Programme (A), ◆ entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch (PAP) dar (M),
<p>3. Einfache Greenfoot-Szenarien</p> <p>(a) Analyse von vorgegebenen Klassen</p> <p>(b) Grundlagen der Vererbung</p> <p>(c) Abfrage von Zuständen</p>	<ul style="list-style-type: none"> ◆ implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), ◆ implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I).

Unterrichtsvorhaben EF-III

Thema: Objektorientierte Analyse, Modellierung und Implementation anhand von komplexen Beispielkontexten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Entwicklung von Spielen oder Simulationen mit Interaktion zwischen unterschiedlichen Objekten in Greenfoot</p> <p>(a) Schrittweise Implementation der Projekte</p> <p>(b) Kontinuierliches Bewegen eines Objekts mit der while-Schleife und der do-while-Schleife</p> <p>(c) Definiertes Bewegen eines Objekts mit der for-Schleife</p> <p>(d) Dokumentation der Klassen des Projekts</p> <p>(e) Kollisionsabfragen</p> <p>(f) Verwendung von Objektreferenzen</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 in Diagrammen 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),
<p>2. Vertiefung der Vererbung</p> <p>(a) Analyse und Erläuterung einer Basis-Version einer Klasse</p> <p>(b) Realisierung von Erweiterungen zur Basisklasse mit Vererbung</p> <p>(c) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung</p> <p>(d) Überschreiben von Methoden</p> <p>(e) abstrakte Oberklassen</p>	<ul style="list-style-type: none"> ◆ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), ◆ ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), ◆ modellieren Klassen unter Verwendung von Vererbung (M), ◆ implementieren Klassen auch unter Nutzung dokumentierter Klassenbibliotheken (I), ◆ testen Programme schrittweise anhand von Beispielen (I),
<p>3. Entwicklung von BlueJ-Projekten mit linearen Datensammlungen fester Größe.</p> <p>(a) Deklaration und Initialisierung von eindimensionalen Felder</p> <p>(b) Zugriff auf die Elemente eines eindimensionalen Feldes</p> <p>(c) Anzahl der Elemente eines eindimensionalen Feldes</p> <p>(d) Darstellung eindimensionaler Felder in Implementationsdiagrammen</p> <p>(e) Eindimensionale Felder als Attribute, Variablen, Parameter und Rückgaben von Methoden</p> <p>(f) Die Klasse Math</p> <p>(g) Die Klasse String</p>	<ul style="list-style-type: none"> ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ modifizieren einfache Algorithmen und Programme (I), ◆ stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), ◆ dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).

Unterrichtsvorhaben EF-IV

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Explorative Erarbeitung eines Sortierverfahrens</p> <p>(a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">◆ beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A),◆ entwerfen einen weiteren Algorithmus zum Sortieren (M),◆ analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).
<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>(f) Analyse des weiteren Sortieralgorithmus (sofern nicht in Sequenz 1 und 2 bereits geschehen)</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) Effizienzbetrachtung zur binären Suche</p>	

Unterrichtsvorhaben EF-V

Thema: Die digitale Welt und die Grundlagen des Datenschutzes

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Informatiker verändern die Welt – Auswirkungen der Informationstechnologie auf Mensch und Gesellschaft</p> <ul style="list-style-type: none">(a) Von analog zu digital(b) Die Binäre Welt(c) Der von-Neumann-Rechner(d) Von der Schrift zum Smartphone(e) Leben in der digitalen Welt(f) Automatisierung(g) Was macht Informatik?	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">◆ stellen ganze Zahlen und Zeichen in Binärcodes dar (D),◆ interpretieren Binärcodes als Zahlen und Zeichen (D),◆ bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),◆ erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),
<p>2. Datenschutz – Der gläserne Mensch</p> <ul style="list-style-type: none">(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“	<ul style="list-style-type: none">◆ nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)

II) Qualifikationsphase

Unterrichtsvorhaben Q1-I

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(a) Analyse der Problemstellung</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p> <p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">◆ analysieren und erläutern objektorientierte Modellierungen (A),◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),◆ modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),◆ ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),◆ modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),◆ implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),◆ nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),◆ wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I),◆ stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),◆ dokumentieren Klassen (D),◆ stellen die Kommunikation zwischen Objekten grafisch dar (D).

Unterrichtsvorhaben Q1-II

Thema: Rekursive Programmierung an mathematischen und grafischen Beispielen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Rekursive und iterative Berechnung von mathematischen Folgen</p> <p>(a) Analyse und Berechnung mathematischer Folgen</p> <p>(b) Implementation rekursiver Methoden zur Berechnung von Folgen</p> <p>(c) Analyse von rekursiven Methoden unter Laufzeit- und Speicherplatzaspekten</p> <p>(d) Analyse einfacher Lindenmayer-Systeme</p> <p>(e) Implementation einfacher Fraktale mithilfe einer Turtle-Grafik</p> <p>(f) Abbruchbedingungen für die Methoden und Aussagen über das Laufzeitverhalten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ analysieren und erläutern Algorithmen und Programme (A), ◆ stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), ◆ entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), ◆ implementieren iterative und rekursive Algorithmen (I), ◆ beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A).

Unterrichtsvorhaben Q1-III

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Queue</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Queue</p> <p>(d) Erstellen eines Implementationsdiagramms für die Anwendung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), ◆ analysieren und erläutern Algorithmen und Programme (A), ◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), ◆ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nicht-lineare Datensammlungen zu (M), ◆ ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), ◆ modifizieren Algorithmen und Programme (I), ◆ implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), ◆ nutzen die Syntax und Semantik einer
<p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse</p>	

<p>Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>	<p>Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</p> <ul style="list-style-type: none"> ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ testen Programme systematisch anhand von Beispielen (I), ◆ stellen lineare und nichtlinear Strukturen grafisch dar und erläutern ihren Aufbau (D).
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List</p>	
<p>4. Vertiefung – Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>	

Unterrichtsvorhaben Q1-IV

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Suchen von Daten in Listen und Arrays</p> <p>(a) Lineare Suche in Listen und in Arrays</p> <p>(b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen</p> <p>(c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ analysieren und erläutern Algorithmen und Programme (A), ◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), ◆ beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), ◆ entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), ◆ modifizieren Algorithmen und Programme (I), ◆ implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), ◆ implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), ◆ nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), ◆ interpretieren Fehlermeldungen und korrigie-
<p>2. Sortieren in Listen und Arrays – Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <p>(a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste</p> <p>(b) Implementierung eines einfachen Sortierverfahrens für ein Feld</p> <p>(c) Entwicklung eines rekursiven Sortierverfahrens für ein Feld (z.B. Sortieren durch Mischen)</p>	
<p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <p>(a) Grafische Veranschaulichung der</p>	

<p>Sortierverfahren</p> <p>(b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarf bei beiden Sortierverfahren</p> <p>(c) Beurteilung der Effizienz der beiden Sortierverfahren</p>	<p>ren den Quellcode (I),</p> <ul style="list-style-type: none"> ◆ testen Programme systematisch anhand von Beispielen (I), ◆ stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).
--	---

Unterrichtsvorhaben Q1-V

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> ▪ Entwicklung von Fragestellungen zur vorhandenen Datenbank ▪ Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> ▪ Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ... FROM, WHERE, AND, OR, NOT) auf einer Tabelle ▪ Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), ◆ analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), ◆ analysieren und erläutern eine Datenbankmodellierung (A), ◆ erläutern die Eigenschaften normalisierter Datenbankschemata (A), ◆ bestimmen Primär- und Sekundärschlüssel (M), ◆ ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), ◆ modifizieren eine Datenbankmodellierung (M), ◆ modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), ◆ bestimmen Primär- und Sekundärschlüssel (M), ◆ überführen Datenbankschemata in vorgegebene Normalformen (M), ◆ verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), ◆ ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),
<p>2. Modellierung von relationalen Datenbanken</p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> ▪ Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms 	

<ul style="list-style-type: none"> ▪ Erläuterung und Modifizierung einer Datenbankmodellierung <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> ▪ Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> ▪ Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation ▪ Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 	<ul style="list-style-type: none"> ◆ stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), ◆ überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).
--	---

Unterrichtsvorhaben Q1-VI

Thema: Sicherheit und Datenschutz in Netzstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), ◆ analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), ◆ untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), ◆ untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A), ◆ nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).
<p>2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht</p>	

Unterrichtsvorhaben Q2-I

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nicht-linearen Datenstrukturen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), ◆ analysieren und erläutern Algorithmen und Programme (A), ◆ beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>	<ul style="list-style-type: none"> ◆ ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), ◆ ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), ◆ modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), ◆ verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), ◆ entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Comparable zur Realisierung einer geeigneten Ordnungsrelation</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>	<ul style="list-style-type: none"> ◆ implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), ◆ modifizieren Algorithmen und Programme (I), ◆ nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), ◆ interpretieren Fehlermeldungen und korrigieren den Quellcode (I), ◆ testen Programme systematisch anhand von Beispielen (I), ◆ stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), ◆ stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>	

Unterrichtsvorhaben Q2-II

Thema: Endliche Automaten und formale Sprachen

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), ◆ analysieren und erläutern Grammatiken regulärer Sprachen (A), ◆ zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), ◆ ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), ◆ entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), ◆ entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), ◆ entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), ◆ entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), ◆ modifizieren Grammatiken regulärer Sprachen (M), ◆ entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), ◆ stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), ◆ ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). ◆ beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).
<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p>	
<p>3. Grenzen endlicher Automaten</p>	

Unterrichtsvorhaben Q2-III

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>(a) Prinzipieller Aufbau einer von Neumann-</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ◆ erläutern die Ausführung eines einfachen

<p>Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>(b) Einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>(c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<p>maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),</p> <ul style="list-style-type: none"> ◆ untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).
<p>2. Grenzen der Automatisierbarkeit</p> <p>(a) Vorstellung des Halteproblems</p> <p>(b) Unlösbarkeit des Halteproblems</p> <p>(c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p>	

Unterrichtsvorhaben Q2-IV

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase