Schulinternes Curriculum im Fach Informatik

Sekundarstufe II (G9) - Abitur 2027 -2029

(Stand: 11/2026)

Inhalt

1 Entscheidungen zum Unterricht 1

1.1 Unterrichtsvorhaben..... 1

1.1.1 Übersicht Unterrichtsvorhaben Einführungsphase..... 1

1.1.2 Übersicht Unterrichtsvorhaben Qualifikationsphase Grundkurs - Leistungskurs.. 12

2 Entscheidungen zu fach- oder unterrichtsübergreifenden Fragen.....24

3 Qualitätssicherung und Evaluation 24

1 Entscheidungen zum Unterricht

1.1 Unterrichtsvorhaben

1.1.1 Übersicht Unterrichtsvorhaben Einführungsphase

Unterrichtsvorhaben:			**Zeitbedarf in Unterrichtsstunden:**
1	Informatik verändert unser Leben		6
2	Die Welt in Klassen und Objekten		9
3	Kennenlernen der Entwicklungsumgebung		6
4	Wir machen einen Plan - Algorithmus		15
5	Verschiedene Arten von Daten		6
6	Klassen und Objekte im Zusammenspiel		12
7	Arrays - Datenzwerge strukturieren		12
8	Alles am richtigen Platz? - Suchen & Sortieren		6
9	Geschichte der Informatik		6
		insgesamt:	78

Unterrichtsvorhaben EF-1: Informatik verändert unser Leben

Zeitbedarf: 6

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Unterrichtssequenzen:

a Unser Kurs/ Unser Unterricht
b Was ist Informatik?
c Informatik all überall

Anregungen zu den Unterrichtssequenzen:

a Abstimmung der Kursregeln, Offenlegung der Erwartungen der Kursteilnehmer bzw. der Erwartungen an die Kursteilnehmer sowie die Kriterien der Leistungsbewertung. Überblick über die zu erwartenden Inhalte

b „In der Informatik geht es genauso wenig um Computer wie in der Astronomie um Teleskope." Dieses Zitat von Dijkstra kann als Ausgangspunkt für eine Diskussion genutzt werden, aus der heraus eine Begriffsdefinition erarbeitet werden soll.
Als Grundlage der weiteren Arbeit sollte an dieser Stelle über die Verwaltung von Daten bzw. Dokumenten innerhalb des pädagogischen Netzwerks des Mariengymnasiums gesprochen werden. Eine Thematisierung von Binärcodes bietet sich an.

c Anhand von möglichen SuS-Beispielen sollen die oben genannten Inhaltsfelder den SuS aufgezeigt werden. Anhand des Beispieles eines Navigationsgerätes oder SmartHomes beispielsweise lässt sich zudem der Grad der Vernetzung der einzelnen Inhaltsfelder erfahren. Ein erster Diskurs im Bereich Informatik, Mensch und Gesellschaft mit Bezügen zu Vor- und Nachteilen bietet sich hier an.

mögliche Vorgehensweisen:

a Kartenabfragen zu den erwarteten Inhalten clustern und zu einer Mindmap zusammenfügen

b Das Thema Binärsystem kann in Form eins Lernens durch Lehren von SuS, die in der SEK I Informatik hatten eingeführt/aufgefrischt werden.

c Die SuS arbeiten anhand vorliegender Inhaltsfelder aus dem Kernlehrplan jeweils ein gesammeltes Beispiel auf, bei dem Sie den Bezug zu den einzelnen Inhaltsfeldern aufzeigen und die Bedeutung kritisch hinterfragen. Es bietet sich an die Ergebnisse in einem Museumsgang zu teilen und eine Diskussion möglicherweise entsprechend in einem Schreibgespräch zu verschriftlichen.

mögliche Werkzeuge

- Kernlehrplan Informatik (Übersicht)
- pädagogische Netzwerke, Office-Tools, …

Unterrichtsvorhaben EF-2: Die Welt in Klassen und Objekten

Zeitbedarf: 9

Inhaltsfelder:

- Algorithmen
- Daten und ihre Strukturierung

Unterrichtssequenzen:

a Ähnlich, aber nicht gleich - Abgrenzung Klasse/Objekt anhand von Attributen
b Datentypen - verschiedene Arten von Attributwerten
c Methoden - was brauchen wir, was bekommen wir?
d Klassen und Objektdiagramme
e Wozu das alles? - Methode von Abbot

Anregungen zu den Unterrichtssequenzen:

a Aus einer Präsentation verschiedener Bilder sollen SuS die Bilder in verschiedene Gruppen unterteilen. In diesen einzelnen Gruppen sollen Gemeinsamkeiten und Unterschiede herausgearbeitet werden. Mit den Ergebnissen werden die Begrifflichkeiten Attribut und Attributwert, Klasse und Objekt verdeutlicht.

b Verschiedene Beispiele aus a ermöglichen eine Unterscheidung der Attributwerte in die einfachen Datentypen Zahl, Text und Wahrheitswert, die bei der Beschreibung einer Klasse mit angegeben werden müssen.

c Nach der Fokussierung auf die verschiedenartigen Objekte einer Klasse, die durch unterschiedliche Attributwerte entstehen wird der Fokus auf Gruppen aus der Präsentation aus a gelenkt, die Fähigkeiten anbieten, wie beispielsweise Tiere oder Maschinen. Hierbei sollen diese Methoden zum einen in Dienste und Anfragen unterteilt werden, zum anderen soll in den Fokus gerückt werden von welchem Datentyp Rückgaben sind oder welche Daten dem Objekt übergeben werden müssen um die Methode auszuführen.

d Als Abschluss werden die Darstellungsform der Objekt- und Klassendiagramme mit Angabe der benötigten Attribute, Attributwerten und Methoden inklusive Rückgabewerte und Übergabewerte eingeführt.

e Die Notwendigkeit der Übersetzung der realen Welt in die Welt der Programme wird in dieser Unterrichtssequenz anhand der Methode von Abbot und anderen Beispielen thematisiert. Der Begriff der Modellierung wird eingeführt.

mögliche Projekte

- Übersetzung verschiedener Märchen wie beispielsweise Hänsel und Gretel in die Sprache der Objekte und Klassen.

mögliche Werkzeuge

- UML-Editoren
- PowerPoint

Unterrichtsvorhaben EF-3: Kennenlernen der Entwicklungsumgebung

Zeitbedarf: 6

Inhaltsfelder:

- Formale Sprachen und Automaten
- Informatiksysteme

Unterrichtssequenzen:

a Bildschirm und Stift - unsere ersten Objekte erstellen und nutzen
b Klassen - Grundaufbau eines Programmes kennenlernen
c Fehlersuche - der Liebling des Informatikers

Anregungen zu den Unterrichtssequenzen:

a Die SuS werden mit den Klassen Stift und Bildschirm vertraut gemacht und lernen die Hilfen und Dokumentationen der Klassen kennen. Sie erkunden durch das Erzeugen des Bildschirmes die Bedeutung der Übergabewerte an den Bildschirm, als Breite, Tiefe und Positionierung. Anschließend nutzen Sie die angebotenen Methoden und erkennen die Unterscheidung in Dienste und Anfragen wieder.

b Die Erstellung eigener Klasse mit der Verwendung von zu erstellenden Objekten ist der nächste Schritt für die SuS, bei dem Sie mit dem Grundaufbau einer Klasse - Deklaration, Initialisierung im Konstruktor, Methoden - vertraut gemacht werden. In dieser Klasse sollen nun Methoden erstellt werden, die das Zeichnen verschiedener Figuren anbieten.

c Anhand vorgegebener Programme sollen die SuS typische Fehlerarten kennenlernen und dabei lernen die angegebenen Fehlerbeschreibungen zu nutzen, um die Fehler zu beheben.

mögliche Projekte

- Überführung der vorgegebenen Klassen Stift und Bildschirm in Klassendiagramme.
- Erstellung eigener fehlerhafter Programme, die von den Mit-SuS behoben werden.

mögliche Werkzeuge

- Java - BlueJ inklusive sum.kern-Bibliothek

Unterrichtsvorhaben EF-4: Wir machen einen Plan - Algorithmus

Zeitbedarf: 15

Inhaltsfelder:

- Algorithmen
- Informatik, Mensch und Gesellschaft

Unterrichtssequenzen:

a Alles nach Plan - Algorithmen im Alltag
b Definition eines Algorithmus
c Volle Kontrolle - Sequenzen, Schleifen und Verzweigungen
d Darstellung von Algorithmen
e Algorithmen entwerfen
f Programmierung von Algorithmen

Anregungen zu den Unterrichtssequenzen:

a Schaffung eines Bewusstseins der Allgegenwärtigkeit von Algorithmen in der Lebenswelt anhand von konkreten Beispielen aus verschiedenen Bereichen: Zähne putzen - Überqueren einer Straße - Bedienung einer Kaffeemaschine - Kochrezepte - Spielregeln - Produktionsvorgänge - Navigationsgeräte …

b Erarbeitung der Kriterien, die einen Algorithmus ausmachen, anhand einiger der obigen Beispiele: Korrektheit, Endlichkeit, Determiniertheit, Determinismus, Terminiertheit

c Erkennen von sich wiederholenden Vorgängen und entscheidungsabhängigen Handlungsabläufen und dahingehende Vereinfachung von Algorithmen durch die Kontrollstrukturen Wiederholung und Verzweigung. Thematisierung der zuvor ausschließlich verwendeten Sequenz.

d Vereinfachung der Verschriftlichung von Algorithmen durch Pseudocode und Struktogramme (auch digital mit dem Struktogrammeditor).

e Entwurf und Visualisierung von Algorithmen unter anderem anhand der Beispiele aus a, aber auch aus anwendungsnahen Gebieten zum Beispiel der Mathematik.

f Umsetzung einiger Struktogramme aus e in Form eines Java-Programmes sowie Überführung von Programmen in Struktogramme.

mögliche Projekte

- Vereinfachung des Quellcodes der Zeichnungen aus Vorhaben EF-3 mit Kontrollstrukturen

- Programmierung eines Dartspiels, dabei auf die Modellierungsphase wert legen mit der Erkenntnis das wir die Interaktion verschiedener Klassen noch nicht kennengelernt haben (mögliche Binnendifferenzierung durch Arbeitsteilung)

mögliche Werkzeuge

- Struktogrammeditor
- Java - BlueJ inklusive sum.kern-Bibliothek

Unterrichtsvorhaben EF-5: Verschiedene Arten von Daten

Zeitbedarf: 6

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:

a Eigenschaften von Klassen & Objekten
b Primitive Datentypen
c Attribute und ihre Umsetzung in Java
d Zufallszahlen
e Vergleich von Daten in Java

Anregungen zu den Unterrichtssequenzen:

a Vertiefung der objektorientierten Denkweise mit Hilfe von alltäglichen Beispielen von Eigenschaften.
b Möglichkeiten der Speicherung von Daten in der digitalen Welt. Vorstellung der wichtigsten Datentypen (wie z. B. boolean, int, double, String, …)
c Das Variablen-Konzept von Java. inkl. Der Deklaration und Initialisierung und der späteren Manipulation der Werte.
d Umsetzung von Variablen in einfachen Anwendungskontexten (Animationen oder einfachen Spielen).
e Der Vergleich von mehreren Werten in Java, z. B. mit den Java-Bezeichnern '<'; '>='; '&&'; '||'

mögliche Projekte

- Schatzsuche
- Sternenhimmel

mögliche Werkzeuge

- Java - BlueJ inklusive sum.kern-Bibliothek

Unterrichtsvorhaben EF-6: Klassen und Objekte im Zusammenspiel

Zeitbedarf: 12

Inhaltsfelder:
- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:
a Klassen und Unterklassen
b Klassendiagramme
c Beziehungen von Klassen & Objekten

Anregungen zu den Unterrichtssequenzen:
a Weiterentwicklung der objektorientierten Sicht durch die Ausschärfung des Klassenbegriffs. Im speziellen kann so die Gruppierung und Hierarchisierung von Klassen erkannt und umgesetzt werden (z. B. mit Hilfe der Unterklasse Figurstift der Klasse Stift).

b Einführung von Klassendiagrammen an einem konkreten Anwendungskontext.
c Vorstellung und Umsetzung der drei Beziehungstypen hat, kennt & ist - am Beispiel eines konkreten Projekts

mögliche Projekte
- CAD
- Billard-Simulation

mögliche Werkzeuge
- Java - BlueJ inklusive sum.kern-Bibliothek

Unterrichtsvorhaben EF-7: Arrays - Datenzwerge strukturieren

Zeitbedarf: 12

Inhaltsfelder:
- Daten und ihre Strukturierung
- Algorithmen
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Unterrichtssequenzen:
a Konsole und Scanner
b for-Schleife
c Die statische Datenstruktur Array
d Arrays in Java-Projekten mit Ein- & Ausgabe

Anregungen zu den Unterrichtssequenzen:
a Einführung in die Basis-Interaktion eines Java-Programms mit Hilfe der Konsole(Ausgabe) und des Scanners(Eingabe).

b Erweiterung der Kenntnisse über Schleifen durch die Wiederholungsschleife (for-Schleife), als nötige Grundlage für das Durchlaufen eines Arrays.

c Aufbau und Struktur eines Arrays mit dem Zugriff über einen Index. Deklaration und Initialisierung eines Arrays in Java. Füllen eines int-Arrays mit Hilfe einer Schleife.

d Fortgeschrittene Projekte zur Umsetzung der gelernten Inhalte (Berechnung im Array, Durchschnittswerte, Minimum, Maximum, …).

mögliche Projekte
- Primzahlengenerator
- Caesar-Verschlüsselung von Buchstaben, Wörtern & Sätzen

mögliche Werkzeuge
- Java - BlueJ
- Caesar-Scheibe

Unterrichtsvorhaben EF-8: Alles am richtigen Platz? - Suchen & Sortieren

Zeitbedarf: 6

Inhaltsfelder:
- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:
a Sinnhaftigkeit einer guten Sortierung
b Such- & Sortierverfahren

Anregungen zu den Unterrichtssequenzen:
a Durch die riesigen Datenmengen im Internet kann das effiziente Suchen als wichtige Voraussetzung der digitalen Verarbeitung von Daten angesehen werden. Diese effiziente Suche ist nur durch eine sinnvolle Sortierung möglich.

b Suchverfahren werden z. B. am Postbotenbeispiel besprochen. Im Anschluss werden einzelne Sortierverfahren vorgestellt und auf ihre Effizienz untersucht und miteinander verglichen. Mögliche Verfahren sind z. B.: Quicksort, Mergesort, Bubblesort, Selectionsort, Shellsort, Insertionsort, ...

mögliche Projekte
- Postbotenbeispiel
- Vortragsreihe Such-& Sortierverfahren

mögliche Werkzeuge
- Präsentationssoftware (Powerpoint, Presenter, Prezi, …)
- Internetshop zur Verdeutlichung der großen Datenmengen

Unterrichtsvorhaben EF-9: Geschichte der Informatik

Zeitbedarf: 6

Inhaltsfelder:
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Unterrichtssequenzen:
a Informatikpersönlichkeiten

Anregungen zu den Unterrichtssequenzen:
a Die Geschichte der Informatik wird über die wichtigsten Personen, die die Wissenschaft Informatik geprägt haben, vorgestellt. Fokus liegt neben der zeitlichen Einordnung auch auf den Erfindungen/ Errungenschaften der einzelnen Persönlichkeiten. Wichtige Persönlichkeiten sind dabei z. B. Ada Lovelace, Alan Turing, Grace Hopper, Steve Jobs, Konrad Zuse, Larry Page & Sergey Brin, Joseph Weizenbaum, ...

mögliche Projekte
- Vortragsreihe

mögliche Werkzeuge
- Präsentationssoftware (Powerpoint, Presenter, Prezi, …)
- Informatikpersönlichkeiten der GI (https://gi.de/aktuelles/projekte/persoenlichkeiten/)

Zuordnung der Kompetenzen in der EF:

Objekte und Klassen	1	2	3	4	5	6	7	8	9
ordnen … … zu									
▫ Attributen, Parametern einfache Datentypen		E			E	V	V	V	
▫ Rückgaben von Methoden einfache Datentypen		E			V	V	V		
▫ Rückgaben von Methoden Objekttypen					V	V	V		
▫ Rückgaben von Methoden lineare Datensammlungen							E		
ordnen … ihren Sichtbarkeitsbereich zu									
▫ Klassen						E			
▫ Attributen						E			
▫ Methoden						E			
stellen … dar									
▫ den Zustand eines Objekts					V	V			
▫ die Kommunikation zwischen Objekten grafisch					V	V			
▫ Klassenbeziehungen in Diagrammen grafisch						E			
▫ Assoziationsbeziehungen in Diagrammen grafisch						E			
▫ Vererbungsbeziehungen in Diagrammen grafisch						E			
dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden		E				V	V		
analysieren und erläutern eine objektorientierte Modellierung		E				V	V		
implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken						V	V	V	

Informatiksysteme	1	2	3	4	5	6	7	8	9
stellen ganze Zahlen und Zeichen in Binärcodes dar	E								
interpretieren Binärcodes als Zahlen und Zeichen	E								
beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner									E
nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen	E				V	V	V	V	V
nutzen das Internet zu Recherche, Datenaustausch und Kommunikation	E							E	V

Formale Sprachen und Automaten	1	2	3	4	5	6	7	8	9
implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik.					V	V	V	V	
interpretieren Fehlermeldungen und korrigieren den Quellcode					V	V	V	V	

Algorithmen	1	2	3	4	5	6	7	8	9
analysieren und erläutern …									
▫ einfache Algorithmen				E	V	V	V	V	
▫ einfache Programme				E	V	V	V	V	
▫ einfache Struktogramme				E	V	V	V	V	
▫ Suchalgorithmen								E	
▫ Sortieralgorithmen								E	
modifizieren einfache …									
▫ Algorithmen				E	V	V	V	V	
▫ Programme				E	V	V	V	V	
▫ Struktogramme				E	V	V	V	V	
entwerfen einfache Algorithmen und stellen sie … dar									
▫ umgangssprachlich				E	V	V	V	V	
▫ grafisch in Form von Struktogrammen				E	V	V	V	V	
implementieren Algorithmen unter Verwendung von …									
▫ Variablen					E	V	V	V	
▫ Wertzuweisungen					E	V	V	V	
▫ Kontrollstrukturen				E	V	V	V	V	
▫ Methodenaufrufen					V	V	V	V	
testen Programme schrittweise anhand von Beispielen			E		V	V	V	V	
wenden Such- und Sortieralgorithmen auf Beispiele an								E	
entwerfen einen weiteren Algorithmus zum Sortieren (M),								E	
beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf								E	

Informatik, Mensch und Gesellschaft	1	2	3	4	5	6	7	8	9
nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und	E			E	V	V	V	V	V
bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen	E								E
erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung									E

1.1.2 Übersicht Unterrichtsvorhaben Qualifikationsphase Grundkurs - Leistungskurs

Qualifikationsphase 1:

Unterrichtsvorhaben:			**Zeitbedarf in Unterrichtsstunden:**
1	Objektorientiertes Java - spielend wiederholt		18
2	Dynamische Datenstrukturen		18
3	Suchen & Sortieren - automatisieren		6
4	Datenbanken - Datenberge strukturieren		24
5	Netzwerkstrukturen & Gefahren im Netz		10
		insgesamt:	76

Qualifikationsphase 2:

Unterrichtsvorhaben:			**Zeitbedarf in Unterrichtsstunden:**
6	Gut abgelegt - nichtlineare Datenstrukturen		24
7	Automaten & Formale Sprachen		20
8	Grenzen der Automatisierbarkeit		12
		insgesamt:	56

Unterrichtsvorhaben Q1-1: Objektorientiertes Java - spielend wiederholt

Zeitbedarf: 18

Inhaltsfelder:
- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:
a Einführung in die Kursarbeit im GK/ LK
b Wiederholung & Erweiterung der OOM
c Wiederholung & Erweiterung der OOP
d (mehrdimensionale) Arrays

Anregungen zu den Unterrichtssequenzen:
a Besprechung der Kursregeln, Vermittlung der Anforderungen des Lehrplans, Vorstellung der Themen für Q1 und der Dokumentation der Bezirksregierung für den jeweiligen Abiturjahrgang.

b Wiederholung der grundlegenden Modellierungskenntnisse zu Struktogrammen, Klassen und Objekten, Beziehungen und Vererbungsbeziehungen zwischen Klassen und Objekten. Erweiterung eines Klassendiagramms zu einem Implementationsdiagramm. Erarbeitung neuer Begriffe wie z. B. abstrakte Klassen, Subtyping, Typecasting und Polymorphie.

c Auffrischung der Java-Kenntnisse mit Hilfe eines Projekts, um die Inhalte der EF zu wiederholen und die neu gelernten Begriffe umzusetzen.

d Vertiefung der bisherigen Kenntnisse über Arrays als Beispiel für eine sinnvolle statische Strukturierung von Werten & Objekten. Übergang von einem eindimensionalen Array zu den zwei- & mehrdimensionalen Arrays in einem geeigneten Anwendungskontext.

mögliche Projekte
- Heldenprojekt
- Vier-Gewinnt
- Schiffe-Versenken

mögliche Werkzeuge
- Java - BlueJ
- Struktogrammeditor
- UML-Editor

Unterrichtsvorhaben Q1-2: Dynamische Datenstrukturen

Zeitbedarf: 18

Inhaltsfelder:
- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:
a Der Einsatz von dynamischen Datenstrukturen
b Schlange
c Stapel
d Liste

Anregungen zu den Unterrichtssequenzen:
a Motivation für den Anwendungskontext dynamische Datenstrukturen durch die Konfrontation der SuS mit der Begrenztheit von statischen Datenstrukturen (z. B. eines Arrays).

b Erarbeitung der Implementation der Queue-Klasse für das Zentralabitur und Programmierung einer Schlangenstruktur (FIFO-Prinzip) in einem geeigneten Anwendungskontext.
c Erarbeitung der Implementation der Stack-Klasse für das Zentralabitur und Programmierung einer Stapelstruktur (LIFO-Prinzip) in einem geeigneten Anwendungskontext.
d Erarbeitung der Implementation der List-Klasse für das Zentralabitur und Programmierung einer Listenstruktur in einem geeigneten Anwendungskontext. Abgrenzung der erlernten dynamischen Datenstrukturen voneinander.

mögliche Projekte
- Verkehrskontrolle

mögliche Werkzeuge
- Java - BlueJ
- Implementationen von Klassen für das Zentralabitur

Unterrichtsvorhaben Q1-3: Suchen & Sortieren - automatisieren

Zeitbedarf: 6

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:

a Vertiefung des Algorithmus-Begriffs
b Rekursionen
c Suchverfahren im Vergleich

Anregungen zu den Unterrichtssequenzen:

a Der Begriff des Algorithmus wird thematisiert und die grundlegenden Eigenschaften besprochen. Darüber hinaus wird besprochen, wie man auch bei komplexen Problemen einen Algorithmus zur Lösung entwickeln kann.

b Eine Grundvoraussetzung für die Interpretation und Entwicklung von Such- und Sortieralgorithmen ist das Verständnis von Rekursionen. Eine rekursive Vorgehensweise wird zunächst an für die SuS verständlichen Beispielen demonstriert bzw. eingeübt und anschließend auf die Programmierung übertragen.

c Die aus der EF bekannten Such-/ Sortierverfahren werden vertieft, indem die Umsetzung in Form von Algorithmen analysiert wird. Es werden Unterschiede bzgl. der Effizienz und iterativen bzw. rekursiven Vorgehensweise beurteilt.

mögliche Projekte

- Fakultätsfunktion - iterativ vs. rekursiv
- Die Türme von Hanoi (Rekursion)

mögliche Werkzeuge

- Java - BlueJ inklusive sum.kern-Bibliothek

Unterrichtsvorhaben Q1-4: Datenbanken - Datenberge strukturieren

Zeitbedarf: 24

Unterrichtssequenzen:

a Speicherung von Daten in Tabellen
b Redundanzen, Anomalien & Inkonsistenzen
c Modellierung von Datenbanken
d SQL - Operationen auf Tabellen
e SQL - Daten Filtern & Verbundoperationen von Tabellen
f Normalisierung
g Datenbankprogrammierung in Java (nur LK)
h Datenschutz(-gesetz)

Anregungen zu den Unterrichtssequenzen:

a In Anknüpfung an die Lebenswelt der SuS werden bekannte Speicherungsformen besprochen und auf ggf. bekannte digitale Speicherungsmethoden eingegangen. Die Ablage von Daten in Tabellen ist den meisten SuS auf Grund der Vorerfahrung mit Tabellenkalkulationsprogrammen (z. B. Excel) bekannt.

b Auftretende Probleme bei den hohen Anforderungen auf Datenbestände (z. B. im Online-Shop) werden thematisiert und so auf verschiedene Datenbankanomalien eingegangen (z. B. Einfüge-/ Lösch- oder Änderungsanomalien).

c Der Nutzen eines sinnvollen Aufbaus von Datenbeständen wird besprochen, um ggf. auftretende Redundanzen zu vermeiden und die SuS lernen mit der Modellierung von Entity-Relationship-Diagrammen (ERM) eine geeignete Darstellungsform von Datenbanken kennen.

d Erste einfach SQL-Befehle werden auf einen bestehenden Datenbestand angewendet und der grundlegende Aufbau einer SQL-Anfrage erarbeitet (SELECT-FROM-WHERE).

e Die SQL-Anfragen werden komplexer, indem Tabellen verbunden und Daten gefiltert werden.

f Probleme mit Anomalien werden erneut aufgegriffen und die Normalisierung als möglich Lösung gegen Inkonsistenzen vorgestellt und für konkrete Tabellen durchgeführt.

g Die erlernten Inhalte werden in selbst geschriebenen Java-Programmen umgesetzt, die entweder auf einen lokalen oder externen SQL-Server zugreifen.

h Die gestiegenen Möglichkeiten werden abschließend kritisch reflektiert. Dabei soll vor allem auf die Gefahren hingewiesen werden, die eine digitale Speicherung von Daten mit sich bringt. Hier wird auf das Datenschutzgesetz eingegangen und konkrete Fälle diskutiert, bei denen eine Verletzung dieses Gesetzes vorliegen könnten.

mögliche Projekte

- Schuldatenbank
- Fallbeispiele in Informatik

mögliche Werkzeuge

- Online-Fitnesscenter
- SQL-Bibliotheken für Programmierung mit Java: MySQL, SQLite, …

Unterrichtsvorhaben Q1-5: Netzwerkstrukturen & Gefahren im Netz

Zeitbedarf: 10

Unterrichtssequenzen:

a Sicherheit in digitalen Netzen
b Grundlagen der Datenübertragung
c Netzwerktopologien
d Routing
e Das OSI-Schichtenmodell
f Aufbau von Rechnernetzen & Heimnetzwerken
g Programmierung von Client-Server-Strukturen (nur LK)
h symmetrische und asymmetrische Verschlüsselung

Anregungen zu den Unterrichtssequenzen:

a In Anknüpfung an die Speicherung von großen Daten wird auch der Austausch von Nachrichten in der digitalen Welt betrachtet.
b Die Grundlage der Datenübertragung wird anhand von klassischen und modernen Varianten der Kommunikation verglichen. Dabei werden auf Chancen und Gefahren von neuen Medien eingegangen (z. B. beim Vergleich von Rauchzeichen & E-Mail). Eine Übertragung über ein Kabel auf dem Strom anliegt oder eben nicht wird als konkretes Beispiel genutzt, um das Morsealphabet zur Kommunikation zu nutzen.
c Die Weiterentwicklung der Vernetzung macht den Wandel von einem Client zu Client zu anderen Netzwerktopologien erforderlich. Geeignete Topologien werden verglichen und auf Vor- & Nachteile hingewiesen.
d Die IP-Adressierung wird als sinnvolle Möglichkeit der Adressierung von einzelnen Clients im WWW vorgestellt und ihre Funktionsweise erarbeitet. Hierbei spiel das Routing eine zentrale Rolle, welches anhand von Beispielen ausprobiert werden kann.
e Einzelne Ebenen der Datenübertragung am Beispiel des OSI-Schichtenmodells besprochen, indem der Ablauf einer Nachricht von einem User zu einem anderen User thematisiert wird.
f Da der reale Aufbau einer Rechnerstruktur nur sehr schwierig erprobt werden kann und die schulischen Voraussetzungen kaum Änderungen der Netzwerkstruktur zulassen, wird die Software Filius eingesetzt, mit deren Hilfe man fiktive Rechnernetze aufbauen kann und die in der Unterrichtsreihe erlernten Inhalte an konkreten Netzwerken ausprobiert werden kann.
g Die zuvor erarbeiteten Netzwerkstrukturen werden in Java Programmen umgesetzt, indem eine Kommunikationsmöglichkeit zwischen verschiedenen Informatiksystemen ermöglicht wird.
h An die in der Einführungsphase gemachten Erfahrungen bezüglich Kryptologie wird angeknüpft. Neben symmetrischen Verschlüsselungsmöglichkeiten wird auch in die asymmetrische Verschlüsselung eingeführt. Abschließend sind die SuS in der Lage mehrere kryptologische Verfahren auf ihre Einsetzbarkeit und Sicherheit kritisch zu prüfen.

mögliche Projekte

- Chat-Programm

mögliche Werkzeuge

- Filius

Unterrichtsvorhaben Q2-6: Gut abgelegt - nichtlineare Datenstrukturen

Zeitbedarf: 24

Inhaltsfelder:
- Daten und ihre Strukturierung
- Algorithmen

Unterrichtssequenzen:
a Ordnung muss sein, Baumstrukturen in verschiedenen Anwendungskontexten
b Binärbaum - mehr als 2 biologische Eltern hat man nicht
c Effizienz beim Suchen durch Baumstrukturen
d Graphen - kürzeste Wege (nur LK)

Anregungen zu den Unterrichtssequenzen:
a Als Einführung kann die Dateistruktur bei Windows-Rechnern betrachtet werden. Anhand dieser Struktur können grundlegende Begrifflichkeiten wie z.B. Grad, Tiefe, Höhe, Blatt etc. rund um Bäume eingeführt werden. Andere Beispiele sind zum Beispiel Termbäume oder Ahnenbäume.

b Mit den Ahnenbäumen gelingt der Übergang auf eine spezielle Baumstruktur, den Binärbaum. Dieser wird als neue nichtlineare Datenstruktur eingeführt und in ein informatisches Modell überführt.
Zur Nutzung dieser Datenstruktur ist der Durchlauf durch einen Baum von entscheidender Bedeutung, so dass die SuS eigenständig Durchlaufstrategien entwickeln sollen, bei denen nach die drei Traversierungsverfahren Pre-, In- und Postorder bei herauskommen sollten. Diese Methoden werden rekursiv implementiert.

c Das Einfügen neuer Knoten in einen Baum, so dass diese bei der Inorder-Traversierung sortiert sind führt zur Klasse BinarySearchTree, deren Mächtigkeit die SuS im Vergleich zu linearen Datenstrukturen erkunden.

d Im LK werden Graphen als eine weitere nichtlineare Datenstruktur aufgeführt, die vor allem durch den Algorithmus von Dijkstra als Anwendungsnah gesehen wird. Thema sind auch die Tiefen- und die Breitensuche und Backtracking.

mögliche Projekte
d Königsberger Brückenproblem

mögliche Werkzeuge
- Java - BlueJ inklusive sum.kern-Bibliothek

Unterrichtsvorhaben Q2-7: Automaten & Formale Sprachen

Zeitbedarf: 20

Inhaltsfelder:

- Automaten und formale Sprachen
- Informatik, Mensch und Gesellschaft

Unterrichtssequenzen:

a Automatisierung im Alltag - Segen oder Fluch?
b Beschreibung und Funktionsweise von Automaten
c Formale Sprachen /reguläre Grammatiken
d Scanner /Parser (nur LK)
e Grenzen der Automatisierbarkeit

Anregungen zu den Unterrichtssequenzen:

a Die SuS identifizieren Formen der Automatisierung im Alltag und entdecken dabei auch das Ersetzen von Arbeitskräften durch Automaten. Anhand dieser selbstgefundenen Beispiele diskutieren Sie die Vor- und Nachteile der fortschreitenden Automatisierung. Mögliche Aspekte dabei sind der Verlust von Arbeitsplätzen, Umgang mit Daten u.vm.

b Anhand eines in Werden entdeckten Kaugummiautomaten oder Parkscheinautomaten wird die formale Beschreibung eines Automaten durch ein 6-Tupel erarbeitet. Die Darstellung der Übergangsfunktion wird dabei sowohl in Tabellenform als auch in Form eines Übergangsgraphen thematisiert.
Anhand dessen wird auch der Begriff des deterministischen Automaten eingeführt und vom nichtdeterministischen Automaten abgegrenzt. Als Spezialfall eines Automaten wird der Akzeptor angeführt.

c Formale Sprachen und reguläre Grammatiken werden in Anlehnung an zuvor erarbeitete deterministische Automaten hergeleitet. Die Überführung eines Automaten in eine reguläre Grammatik und umgekehrt wird geübt.
Das Vorkommen solcher formalen Sprachen wird in Programmiersprachen und auch unserer Sprache erkannt und verdeutlicht. Dabei wird im LK der Begriff der kontextfreien Grammatik thematisiert.

d Im LK werden zusätzlich Scanner und Parser thematisiert. Dies kann anhand des Ablaufes eines Compilers bzw. Interpreters eingeführt werden.

e Die Diskussion aus der Unterrichtssequenz a wird nun um den Aspekt erweitert, inwiefern die Automatisierung technisch auf Grenzen stößt.

mögliche Projekte

a Zu den Aspekten können Referate verteilt werden, die die entsprechenden Aspekte vertiefen und im Zuge von Unterrichtssequenz e nochmals zur Sprache gebracht werden.
b Ausflug in die Fußgängerzone von Werden und Testen von Automaten mit dem Auftrag eine formale Beschreibung eines Automaten zu erarbeiten.

mögliche Werkzeuge

- JFLAP

Unterrichtsvorhaben Q2-8: Grenzen der Automatisierbarkeit

Zeitbedarf: 12

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Unterrichtssequenzen:

a Aufbau von Informatiksystemen

b Grenzen der Automatisierbarkeit (nur LK)

Anregungen zu den Unterrichtssequenzen:

a Erörtert wird der prinzipielle Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher.
Einige maschinennahe Befehle und ihre Repräsentation in einem Binärcode können thematisiert werden.

b Technische und physikalische Grenzen der Automatisierbarkeit werden thematisiert, ebenso wird nochmals eine Diskussion über ethische und moralische Grenzen angestoßen.

mögliche Werkzeuge

- Java - BlueJ inklusive sum.kern-Bibliothek

Zuordnung der Kompetenzen in der Qualifikationsphase:

Objekte und Klassen	nur LK	1	2	3	4	5	6	7	8
ordnen … Datentypen, Objekttypen sowie Datensammlungen zu									
▪ Attributen		V	V	V	V	V			
▪ Parametern		V	V	V	V	V			
▪ Rückgaben von Methoden		V	V	V	V	V			
ordnen … ihren Sichtbarkeitsbereich zu									
▪ Klassen		V	V	V	V	V			
▪ Attributen		V	V	V	V	V			
▪ Methoden		V	V	V	V	V			
stellen … dar									
▪ lineare Strukturen grafisch		V	V	V					
▪ nichtlineare Strukturen graifisch							E		
▪ die Kommunikation zwischen Objekten		V	V	V		V			
▪ Klassen und ihre Beziehungen in Diagrammen		V	V	V		V			
ermitteln … von Objekten									
▪ Eigenschaften		V	V	V		V			
▪ Operationen		V	V	V		V	V		
▪ Beziehungen		V	V	V		V	V		
modellieren Klassen mit …									
▪ Attributen		V	V	V		V	V		
▪ Methoden		V	V	V		V	V		
▪ Assoziationsbedingungen (inkl. Multipliziäten)		E	V	V		V	V		
modellieren … unter Verwendung von Vererbung durch Spezialisiern und Generalisieren									
▪ abstrakte Klassen		E							
▪ nichtabstrakte Klassen		V	V	V		V			
verwenden Möglichkeiten der Polymorphie		E							
analysieren und erläutern objektorientierte Modellierung		V	V	V		V	V		
implementieren Klassen in einer Programmiersprache		V	V	V	V	V	V		
dokumentieren Klassen		V	V	V	V	V	V		

Datenbanken	nur LK	1	2	3	4	5	6	7	8
ermitteln …									
▪ Entitäten					E				
▪ Attribute					E				
▪ Relationen					E				
▪ Kardinalitäten					E				
stellen … in einem Entity-Relationship-Diagramm dar									
▪ Entitäten					E				
▪ Attribute					E				
▪ Beziehungen (mit Kardinalitäten)					E				
bestimmen …									
▪ Primärschlüssel					E				
▪ Sekundärschlüssel					E				
▪ Fremdschlüssel					E				
überführen Datenbankschemata …									
▪ auf vorgegebene Normalisierungseigenschaften					E				
▪ in die 1. bis 3. Normalform					E				
modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema					E				
analysieren und erläutern eine Datenbankmodellierung					E				
erläutern die Eigenschaften normalisierter Datenbankschemata					E				
modifizieren eine Datenbankmodellierung	x				E				
implementieren ein relationales Datenbankschema als Datenbank	x				E				

Algorithmen	nur LK	1	2	3	4	5	6	7	8
analysieren und erläutern …									
▪ Algorithmen		V	V	V	V	V	V		
▪ Programme		V	V	V	V	V	V		
modifizieren …									
▪ Algorithmen		V	V	V	V	V	V		
▪ Programme		V	V	V	V	V	V		
stellen iterative & rekursive Algorithmen … dar									
▪ umgangssprachlich		V	V	V	V	V			
▪ grafisch		V	V	V	V	V			
entwickeln iterative und rekursive Algorithmen unter Nutzung von …									
▪ Modularisierung			E	V	V	V	V		
▪ Teilen und Herrschen		V	V	V	V	V	V		
▪ dynamischen Datenstrukturen			E	V	V	V	V		
erläutern Operationen dynamischer … Datenstrukturen									
▪ linearer			E				V		
▪ nichtlinearer							E		
testen Programme systematisch anhand von Beispielen		V	V	V	V	V			
implementieren und erläutern iterative und rekursive Such- und Sortierverfahren				E					
beurteilen die Effizienz von Algorithmen anhand des Speicherbedarfs, Zahl der Operationen				E			V		
ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen					E				
entwickeln iterative und rekursive Algorithmen unter Nutzung von Backtracking	x			E			E		
testen Programme systematisch mithilfe von Testanwendungen	x	V	V	V	V	V	V		
implementieren Operationen dynamischer Datenstrukturen	x		E	V	V	V	V		
implementieren uiterative und rekursive Such- und Sortierverfahren	x			E					
erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexität	x			E					
erläutern das Prinzip der Nebenläufigkeit	x					E			
analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation	x					E			
entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikatio	x					E			

Formale Sprachen und Automaten	nur LK	1	2	3	4	5	6	7	8
nutzen die Syntax und Semantik einer Programmiersprache …									
▪ bei der Implementierung		V	V	V	V	V	V		
▪ zur Analyse von Programmen		V	V	V	V	V	V		
beurteilen die … von Programmen									
▪ syntaktische Korrektheit		V	V	V	V	V	V		
▪ Funktionalität		V	V	V	V	V	V		
analysieren und erläutern …									
▪ die Syntax und Semantik einer Datenbankabfrage					E				
▪ die Eigenschaften endlicher Automaten (z. B. bei Eingaben)								E	
▪ Grammatiken regulärer Sprachen								E	
stellen endliche Automaten … dar und überführen Sie in die andere Darstellungsform									
▪ Tabellen								E	
▪ Graphen								E	
ermitteln die …									
▪ Sprache, die ein endlicher Automat akzeptiert								E	
▪ die formale Sprache, die durch eine Grammatik erzeugt wird								E	
entwickeln …									
▪ zur Grammatik einer regulären Sprache einen endlichen Automaten								E	
▪ zu einer regulären Sprache eine Grammatik die die Sprache erzeugt								E	
▪ zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik								E	
Interpretieren Fehlermeldungen und korrigieren den Quellcode		V	V	V	V	V			
verwenden die Syntax und Semantik einer Datenbankabfragesprache					E				
entwickeln und modifizieren endliche Automaten zu einer Problemstellung								E	
modifizieren Grammatiken regulärer Sprachen								E	
beschreiben den Zusammenhang zwischen Automaten und Grammatiken								E	
zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf								E	

Formale Sprachen und Automaten	nur LK	1	2	3	4	5	6	7	8
analysieren und erläutern die Eigenschaften von Kellerautomaten (z. B. bei Eingaben)	x							E	
ermitteln die Sprache, die ein Kellerautomat akzeptiert	x							E	
entwickeln und modifizieren Kellerautomaten zu einer Problemstellung	x							E	
entwickeln zur Grammatik einer kontextfreien Sprache einen zugehörigen...									
▪ endlichen Automaten	x							E	
▪ Kellerautomaten	x							E	
analysieren und erläutern Grammatiken kontextfreier Sprachen	x							E	
modifizieren Grammatiken kontextfreier Sprachen	x							E	
entwickeln zu einer kontextfreien Sprache eine Grammatik, die die Sprache erzeugt	x							E	
modellieren und Implementieren ... zu einer gegebenen regulären Sprache									
▪ Scanner	x							E	
▪ Parser	x							E	
▪ Interpreter	x							E	

Informatik, Mensch und Gesellschaft	nur LK	1	2	3	4	5	6	7	8
untersuchen und bewerten anhand von Fallbeispielen ...									
▪ Auswirkungen des Einsatzes von Informatiksystemen					E	V			
▪ Aspekte der Sicherheit von Informatiksystemen						E			
▪ Aspekte der Sicherheit des Datenschutzes					E				
▪ Aspekte der Sicherheit des Urheberrechts					E				
untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hisichtlich ...									
▪ rechtlicher Vorgaben					E				
▪ ethischer Aspekte					E				
▪ gesellschaftlicher Werte					E				
untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen									E

Informatiksysteme	nur LK	1	2	3	4	5	6	7	8
erläutern ...									
▪ die Ausführung eines einfachen maschinennahen Programms									E
▪ die Datenspeicherung auf einer "Von-Neumann-Atchitektur"									V
beschreiben und erläutern ...									
▪ Netzwerk-Topologien						E			
▪ Client-Server-Struktur						E			
▪ Protokolle						E			
▪ ein Schichtenmodell in Netzwerken						E			
nutzen Informatiksysteme und das Internet zur ...									
▪ Erschließung fachlicher Inhalte		V	V	V	V	V			
▪ Aufbreitung fachlicher Inhalte		V	V	V	V	V			
▪ Präsentation fachlicher Inhalte		V	V	V	V	V			
▪ strukturierten Verwaltung von Dateien		V	V	V	V	V			
wenden eine didaktisch orientierte Entwicklungsumgebung ... an									
▪ zur Demonstration		V	V	V	V	V			
▪ zum Entwurf		V	V	V	V	V			
▪ zur Implementierung		V	V	V	V	V			
▪ zum Test von Informatiksystemen		V	V	V	V	V			
erläutern ... von Datenbanksystemen unter dem Aspekt der sicheren Nutzung									
▪ Eigenschaften					E				
▪ Funktionsweisen					E				
▪ Aufbau					E				
analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren						E			
analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk	x					E			
entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk	x					E			
entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem	x	V	V	V	V	V			

2 Entscheidungen zu fach- oder unterrichtsübergreifenden Fragen

In der gymnasialen Oberstufe erweitern und vertiefen die Schülerinnen und Schüler ihr Informatikwissen. Kernthemen sind hier die formale Modellierung (z. B. Objektorientierung, Datenbanken, formale Sprachen) sowie die Entwicklung komplexerer Algorithmen. Ein weiteres Ziel ist die Vermittlung abstrakter Konzepte der theoretischen Informatik (z. B. endliche Automaten, Komplexität). Der Unterricht fördert selbstständiges Arbeiten: Die Schülerinnen und Schüler planen und realisieren eigene Software- bzw. Hardwareprojekte, wenden vernetzte Informationssysteme an und reflektieren ethische und gesellschaftliche Aspekte der Digitalisierung. Methodisch steht die Anwendung mathematisch-logischer Fähigkeiten im Vordergrund, ebenso wie fachgerechte Problemlösestrategien. Ziele sind außerdem die Vorbereitung auf die digitalisierte Arbeitswelt und – im Gymnasialbereich – die gezielte Abiturvorbereitung im Fach Informatik.

Der Informatikunterricht der Sekundarstufe II legt den Fokus auf vertiefte Inhalte und die Förderung höherer Denkfähigkeiten. Er verwendet präzise Fachbegriffe und nennt konkrete Themenfelder, wie sie in den Oberstufen-Kernlehrplänen vorgesehen sind. Die selbständigen Projekte und Problemlösungsstrategien werden den älteren Schülerinnen und Schülern gerecht, die zunehmend selbstverantwortlich lernen.

3 Qualitätssicherung und Evaluation

Durch die Möglichkeit parallel gestellter Klassenarbeiten oder Klausuren, durch Diskussion der Aufgabenstellungen und Ergebnisse von Klassenarbeiten und Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Lernstandserhebung, zentraler Klausur am Ende der Einführungsphase und Zentralabitur wird ein hohes Maß an fachlicher Qualitätssicherung erreicht. Darüber hinaus dient die Vorlage von Belegexemplaren der Leistungsüberprüfungen mit Erwartungshorizonten an die Fachkoordination der Sicherung der fachlichen Qualität. In der Fachkonferenz diskutieren Fachlehrkräfte, Eltern- und Schülervertreter über Möglichkeiten der Unterrichtsentwicklung, Einführung von Lehr- und Lernmitteln und Veränderungen im schulinternen Curriculum. Nach Absprache innerhalb der Fachgruppe wird erkannter Fortbildungsbedarf zusammengetragen und an die Fortbildungskoordination weitergeben. Individuelle Fortbildungen werden in der Fachgruppe vorgestellt und gemeinsam zur Unterrichtsentwicklung genutzt. Im Sinne einer Feedbackkultur können freiwillige kollegiale Unterrichtshospitationen und Evaluationsfragebögen eingesetzt werden und zur weiteren Professionalisierung beitragen.