Goerdeler-Gymnasium Paderborn

Schulinternes Curriculum für das Fach Informatik in der Sekundarstufe II (EF, Q1, Q2) ab Abiturjahrgang 2017

auf Grundlage der Richtlinien und Kernlehrpläne sowie der fachlichen Vorgaben zum Zentralabitur in NRW

Stand: April 2015, Fachkonferenzbeschluss

INHALT

1Die Fachgruppe Informatik des Goerdeler-Gymnasium Paderborn... 3

2 Entscheidungen zum Unterricht 5

2.1 Unterrichtsvorhaben 5
2.1.1 Übersichtsraster Unterrichtsvorhaben 6
2.1.2 Konkretisierte Unterrichtsvorhaben 9
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.. 23
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung 24

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen . 26

4 Qualitätssicherung und Evaluation 27

1 Die Fachgruppe Informatik des Goerdeler-Gymnasium Paderborn

Beim Goerdeler-Gymnasium handelt es sich um eine dreizügige Schule am Rande von Paderborn mit zurzeit ca. 800 Schülerinnen und Schülern, und ca. 70 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule sind die umliegenden Stadtteile und Eingemeindungen.
Das Fach Informatik wird am Goerdeler-Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweistündig unterrichtet.

In der Sekundarstufe II bietet das Goerdeler-Gymnasium für die eigenen Schülerinnen und Schüler in allen Jahrgangsstufen jeweils einen Grundkurs in Informatik an.

Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei zusätzlich eine didaktische Bibliothek zum Einsatz, welche das Erlernen der objektorientierten Denkweisen erleichtert.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik des Goerdeler-Gymnasiums aus zwei Lehrkräften, denen drei Computerräume mit jeweils 16 Schüler-Computerarbeitsplätzen und ein Selbstlernzentrum mit 8 Plätzen zur Verfügung stehen. Alle Arbeitsplätze sind an das schulinterne Rechnernetz „Lernstatt Paderborn“ angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der drei Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

2.1.1 Übersichtsraster Unterrichtsvorhaben

Einführungsphase

UV	Thema, Schwerpunkt, Beispiele	Std.
E-I	*Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik* *a) **Die Inhaltsfelder des Fachs Informatik (am Beispiel eines Navigationsgeräts)*** *(1) Daten und ihre Strukturierung; (2) Algorithmen; (3) Formale Sprachen und Automaten;* *(4) Informatiksysteme; (5) Informatik, Mensch und Gesellschaft* *b) **Der kompetente Umgang mit dem Schulnetzwerk*** *(1) Erstellen und Anlegen von Ordnerstrukturen; (2) Sortieren von Dateien und Ordnern;* *(3) Eingabe von Befehlen über Eingabeaufforderung; (4) Einzelrechner und Netzwerk;* *(5) Sicherheit und Datenschutz*	6 2W
E-II	*Grundlagen der objektorientierten Analyse, Modellierung und Implementierung* (Objekte, Klassen, Syntax und Semantik einer Programmiersprache (Java)) *a) Begriffe der Objektorientierung (Objekt-/Klassendiagramme)* *b) Grundlegender Aufbau einer Java-Klasse (Deklaration, Initialisierung, Methodenaufruf)* *c) Analyse und Implementierung von Quelltexten (lineare Sequenzen)*	8 3W
E-III	*Algorithmische Grundstrukturen in Java* (Analyse, Entwurf und Implementierung einfacher Algorithmen) *a) Erarbeitung von Kontrollstrukturen (Schleifen, Verzweigungen)* *b) Einsatzmöglichkeiten von Variablen (lokale Variable, Parameterübergabe, Rückgabewerte)*	18 6W
E-IV	*Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung* *a) binäres Zahlensystem (Darstellung ganzzahliger Zahlen)* *b) binäre Codierung von Zeichen (ASCII, UniCode, …)* *c) Grundlegender Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur*	8 3W
E-V	*Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele* *a) Von der Anforderungsbeschreibung zum Entwurfsdiagramm* *b) Vom Entwurfsdiagramm zum Implementationsdiagramm (Klassendokumentaion, Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen)* *c) Konzept der Vererbung (Spezialisierung und Verallgemeinerung, Überschreiben von Methoden)*	18 6W
E-VI	*Such- und Sortieralgorithmen* (Analyse, Entwurf und Implementierung einfacher Algorithmen) *a) Systematisierung von Strategien, Notation im Pseudocode, mindestens 3 Verfahren ohne rekursive Verfahren: Sortieren durch Einfügen (Insertion Sort), durch Vertauschen (Bubble Sort) und durch Auswählen (Selection Sort)* *b) Effizienzbetrachtungen: Rechenzeit (Variation der Sortierverfahren), Speicherplatzbedarf (Feld (Array) als Datensammlung)* *c) Binäre Suche auf sortierten Daten*	9 3W

UV	Thema, Schwerpunkt, Beispiele	Std.
E-VII	*Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?* (Wirkungen der Automatisierung, Geschichte der automatischen Datenverarbeitung, Digitalisierung) *a) Kleingruppenarbeit mit (Plakat-) Präsentationen zu „Vom Morsen zum Digitalcomputer", „von Caesar zur Enigma", „Stellenwertsysteme", „ASCII, RGB und mehr", „Veränderungen der Arbeitswelt und Datenschutz"* *b) Vertiefung des Themas Datenschutz, Diskussion und Bewertung von Fallbeispielen*	10 4W

Qualifikationsphase

UV	Thema, Schwerpunkt, Beispiele	Std.
Q1 I	Wiederholung der objektorientierten Modellierung und Programmierung (abstrakte Oberklasse als Generalisierung verschiedener Unterklassen, grafische Darstellung der Kommunikation zwischen zwei ausgewählten Objekten: Sequenzdiagramm)	12 4W
Q1 II	Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen Schlange (Queue), Stapel (Stack), Liste (List) Modellierung mit Entwurfs- und Implementationsdiagrammen (Patientenwarteschlange, Kisten stapeln, Abfahrtslauf, Skispringen, Terme in Postfix-Notation, Rangierbahnhof, Zufahrtsregelung mit Ampeln, Karteiverwaltung)	24 8W
Q1 III	Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen (Liste, Feld) Suchen (linear, binär): Karteiverwaltung, Bundesjugendspiele Entwicklung und Implementierung iterativer und rekursiver Sortierverfahren Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen" und „Quicksort" auf linearen Listen	20 7W
Q1 IV	Automaten und formale Sprachen a)Vom realen Automaten zum formalen Modell endlicher Automaten (NEA, DEA) b) Grammatiken regulärer Sprachen und endliche Automaten c) Anwendungen, die durch endliche Automaten oder reguläre Grammatiken beschrieben werden können	20 7W

UV	Thema, Schwerpunkt, Beispiele	Std.
Q2 I	Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen Baumstrukturen (am Beispiel: Termbaum, Ahnenbaum, Suchbaum, Entscheidungsbaum, Codierungsbaum) Informatikerbaum als binärer Baum, dann als Suchbaum (Erweiterung; Huffman-Codierung, Buchindex)	20 7W
Q2 II	Aufbau von und Kommunikation in Netzwerken Sicherheit und Datenschutz in Netzstrukturen, Topologie von Netzwerken, Client-Server-Strukturen, TCP/IP-Schichtenmodell, symmetrische, asymmetrische kryptographische Verfahren, Fallbeispiele zur	16 6W

UV	Thema, Schwerpunkt, Beispiele	Std.
	Datenschutzproblematik und zum Urheberrecht	
Q2 III	Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten Grundbegriffe, SQL-Abfragen, ER-Diagramme, Redundanz, Konsistenz, Normalformen 1. bis 3. (Videocenter, Schulbuchausleihe, Fahrradverleih, Reederei, Buchungssystem, Schulverwaltung)	20 7W
Q2 IV	Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase	6 2W

2.1.2 Konkretisierte Unterrichtsvorhaben

EINFÜHRUNGSPHASE

Sequenzierung des Unterrichtsvorhabens E-I:

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

Sequenzierung des Unterrichtsvorhabens E-II:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Identifikation von Objekten und Klassen (a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt. (c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.	Die Schülerinnen und Schüler - 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).	***Kapitel 2 Einführung in die Objektorientierung*** 2.1 Objektorientierte Modellierung
		Kapitel 2 Einführung in die Objektorientierung 2.2 Von der Realität zu Objekten; Von den Objekten zu Klassen, Klassendokumentation; Objekte inspizieren; Methoden aufrufen; Objektidentität und Objektzustand; z.B. Greenfoot-Szenario „Planetenerkundung“
2. Analyse von Objekten und Klassen (a) Schritte der objektorientierten Analyse, Modellierung und Implementation (b) Analyse und Erprobung der Objekte in einer geeigneten Entwicklungsumgebung		***Kapitel 2 Einführung in die Objektorientierung***

3. Implementierung einfacher Aktionen in Greenfoot (a) Quelltext einer Java-Klasse (b) Implementation eigener Methoden, Dokumentation mit JavaDoc (c) Programme übersetzen (Aufgabe des Compilers) und testen		2.3 Programmierung in Greenfoot oder GLOOP oder „Shapes and Sprites“ Methoden schreiben Programme übersetzen und testen

Sequenzierung des Unterrichtsvorhabens E-III:

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), - 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).	***Kapitel 3 Algorithmen*** 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 Algorithmen entwickeln
2. Variablen und Methoden (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstruktoren (d) Realisierung von Attributen		***Kapitel 4 Variablen und Methoden*** 4.1 lokale Variablen 4.2 Methoden 4.3 Attribute

Sequenzierung des Unterrichtsvorhabens E-IV:

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)	***Exkurs „Analog und Digital“*** ***Exkurs „Binäre Welt“***
		Exkurs „Arbeitsweise eines Computers“

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)	
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		

Sequenzierung des Unterrichtsvorhabens E-V:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
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 **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	Die Schülerinnen und Schüler - 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),	***Kapitel 6 Klassenentwurf*** 6.1. Von der Realität zum Programm 6.2. Objekte identifizieren 6.3. Klassen und Beziehungen entwerfen ***Kapitel 6 Klassenentwurf*** 6.4 Klassen und Beziehungen implementieren 6.5 Vererbung ***Kapitel 6 Klassenentwurf*** 6.4 Klassen und Beziehungen implementieren 6.5 Vererbung ***Kapitel 6 Klassenentwurf*** 6.5 Vererbung ***Kapitel 8 Softwareprojekte*** 8.1 Softwareentwicklung 8.2 Oberflächen

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.	- 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 Vererbungs-beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)	
4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert		
5. Softwareprojekt (a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) (b) Erweiterung des Spiels um weitere Funktionalitäten (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche (d) (arbeitsteilige) Implementation des Spiels		

Sequenzierung des Unterrichtsvorhabens E-VI:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Modellierung und Implementation von Datenansammlungen (a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld	Die Schülerinnen und Schüler - 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) - ordnen Attributen lineare Datenansammlungen zu (M)	***Kapitel 7 Sortieren und Suchen auf Feldern*** 7.1 Das Feld – Eine Sammlung von Daten
2. Explorative Erarbeitung von Suchverfahren (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion. (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.		***Kapitel 7 Sortieren und Suchen auf Feldern*** Projekteinstieg 1: Suchen 7.2 Suchen mit System Lineare Suche Binäre Suche Hashing
		Kapitel 7 Sortieren und Suchen auf Feldern Projekteinstieg 2: Sortieren

3. Systematisierung von Algorithmen und Effizienzbetrachtungen (a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs (e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)		7.3 Ordnung ist das halbe Leben!? – Sortieren Sortieren Selection Sort Insertion Sort Bubble Sort

Sequenzierung des Unterrichtsvorhabens E-VII:

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 - 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)	***Exkurs „Geschichte der Informatik“*** ***Exkurs „Informatik und Gesellschaft“***
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.		

Hinweise:
- Die Texte sind aus dem neuen Lehrwerk Informatik 1 des Schöningh Verlags und dessen Vorwort bzw. dessen Kommentierungen entlehnt.

- Das vorliegende Übersichtsraster und die Sequenzierungen sollen am Ende des Schuljahrs 2014/2015 evaluiert und überarbeitet werden.

QUALIFIKATIONSPHASE

Sequenzierung des Unterrichtsvorhabens Q1-I:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung a) Sichtweise der objektorientierten Informatik auf die Welt b) OOP als informatikspezifische Modellierung der Realität c) Schritte der Softwareentwicklung	Die Schülerinnen und Schüler … - analysieren und erläutern objektorientierte Modellierungen (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), - 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), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes 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).	2.1 Die Welt ist voller Objekte Projekteinstieg: Klassenentwurf – step by step
2. Erweiterung der objektorientierten Programmierung a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm b) Objektkommunikation im Sequenzdiagramm c) Klassendokumentation d) Umsetzung von Teilen der Modellierung		2.2 Gut geplant – Klassenentwurf 2.3 Vererbungshierarchien nutzen
3. Mensch und Technik a) Verantwortung von Informatikern b) Automatisierung des Alltags durch Informatik		Die digitale Welt 001 - Mensch und Technik
4. Übung und Vertiefung der OOM / OOP		Prüfungsvorbereitung

Sequenzierung des Unterrichtsvorhabens Q1-II:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Die Datenstruktur **Feld** a) Erarbeitung der Anforderungen an eine Datenstruktur b) Wiederholung der Datenstruktur Array, Eigenschaften der	Die Schülerinnen und Schüler… - erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A)	**Kapitel:** - Anforderungen an eine Datenstruktur - Speichern mit Struktur *neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10,*

Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays c) Modellierung und Implementierung von Anwendungen	- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M) - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)	*eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)*
2. Die Datenstruktur **Schlange** a) Modellierung und Implementierung der Verknüpfung von Objekten b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS. c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden d) Funktionalität der Schlange unter Verwendung der Klasse *Queue*; Erschließen der Standardoperationen e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse *Queue*	- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) - testen und korrigieren Computerprogramme (I) - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D) - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M) - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) - dokumentieren Klassen (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	**Kapitel:** - Wer zuerst kommt - Objekte miteinander verketten - Verwaltung und Inhalt - Funktionen der Queue **Aufgaben:** - Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist. - Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus dem Verkehr gewunken. Es werden so lange Fahrzeuge kontrolliert, bis eine gewissen Menge an Verstößen vorliegt oder Autos kontrolliert wurden.
3. Die Datenstruktur **Stapel** a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden b) Funktionalität der Klasse Stapel unter Verwendung der Klasse *Stack* c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse *Queue* d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der		**Kapitel:** - Daten gut abgelegt – Stapel - Funktionen der Datenstruktur Stapel **Aufgaben:** - Standardoperationen/Basiskompetenz (Stapel Münzen/ CDs) zur Umsetzung der gegebenen Funktionen der Klasse *Stack* - Biber und Teller Es gibt große und kleine Biber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber

Klassen *Queue*, *Stack* und *Array (Palindrom)*)		passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl Queue als auch Stack verwendet werden. - Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.
4. Die Datenstruktur **Liste** a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu *Stack*/*Queue*, zusätzliche Fähigkeiten der Klasse *List*) b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse *List* c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse *List* d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List)		**Kapitel:** - Flexibel für alle Fälle – (die) lineare Liste - Funktionen der Datenstruktur Liste **Aufgaben:** - LEDs - Textzeilen verarbeiten
5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen		Prüfungsvorbereitung
		Projekteinstieg Heldenspiel *Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.*

Sequenzierung des Unterrichtsvorhabens Q1-III:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Eigenschaften von Algorithmen a) Qualitätseigenschaften von Algorithmen	Die Schülerinnen und Schüler - analysieren und erläutern Algorithmen und Programme (A),	4.1 Ohne Algorithmen läuft nichts

b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“ c) Analyse und Entwicklung von rekursiven Algorithmen	- modifizieren Algorithmen und Programme (I), - 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),	4.2 Teile die Arbeit – rekursive Algorithmen
2. Suchen in Listen und Arrays a) Lineare Suche in Listen und Arrays b) Binäre Suche in einem Array c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf	- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - testen Programme systematisch anhand von Beispielen (I). - implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),	4.3 Suchen – iterativ und rekursiv
3. Sortieren auf Listen und Arrays a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf	- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - 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),	4.4 Sortieren – iterativ und rekursiv

Sequenzierung des Unterrichtsvorhabens Q1-IV:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Endliche Automaten a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus c) Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere	Die Schülerinnen und Schüler - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A) - ermitteln die Sprache, die ein endlicher Automat akzeptiert (D) - entwickeln und modifizieren zu einer Problemstellung endlicher Automaten (M) - stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D) - entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M) - analysieren und erläutern Grammatiken regulärer Sprachen (A)	Vom realen Automaten zum Modell Projekteinstieg: Schatzsuche Der Mealy-Automat Der erkennende endliche Automat Wort und Sprache
2. Grammatiken regulärer Sprachen		Grammatiken regulärer

a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften) b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.	- modifizieren Grammatiken regulärer Sprachen (M) - entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M) - entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M) - beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D) - zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf	Sprachen
3. Übungen und Vertiefungen Verwendung endlicher Automaten und Grammatiken regulärer Sprachen		Prüfungsvorbereitung
Projekteinstieg *Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben*		

Sequenzierung des Unterrichtsvorhabens Q2-I:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Aufbau von Baumstrukturen und Grundbegriffe a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten	Die Schülerinnen und Schüler - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), - analysieren und erläutern Algorithmen und Programme (A), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - 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), - 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),	6.1 Spielen mit Struktur – Baumstrukturen Projekteinstieg 1: Spielbäume
2. Binäre Bäume a) rekursiver Aufbau eines binären Baums b) Traversierungen (pre-, in-, postorder) c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm) d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)		6.2 Zwei Nachfolger sind genug! - Binäre Bäume Implementation des Projekts Ahnenbaum
3. Binäre Suchbäume a) Prinzip des binären Suchbaums, Ordnungsrelation b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe) c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)		6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume Projekteinstieg 2: Binäre Suchbäume Implementation des Projekts Benutzerverwaltung
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		Prüfungsvorbereitung

Sequenzierung des Unterrichtsvorhabens Q2-II:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Technische Kommunikation als Fortführung natürlicher Kommunikation a) Kommunikation im Shannon-Weaver-Modell b) Kriterien von technischen Kommunikationsarten c) Die Geschichte der technischen Kommunikation	Die Schülerinnen und Schüler - beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).	7.1 Menschen kommunizieren – ohne und mit Technik Projekteinstieg: Kommunikation im Wilden Westen
2. Aufbau von Netzwerken und Kommunikationsregeln a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen		7.2 Ohne Protokoll läuft nichts - Netzwerke
3. Aufgabenteilung in Netzwerken durch Server und Client a) Aufbau und Aufgaben der Client-Server-Struktur (b) Protokolle zwischen Client und Server		7.3 Einer für alle – Client-Server-Struktur
4. Kryptologie a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA) b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand		Die digitale Welt 100 – Kryptologie
5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken		Prüfungsvorbereitung

Sequenzierung des Unterrichtsvorhabens Q2-III:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen b) SQL-Abfragen - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) - Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY,ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (…), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke) c) Vertiefung an einem weiteren Datenbankbeispiel	Die Schülerinnen und Schüler - 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), - überführen Datenbankschemata in vorgegebene Normalformen (M), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einen Datenbanksystem zu extrahieren (I), - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	**Kapitel 8 Datenbanken** 8.1 Wissen speichern und verwalten – Datenbanksysteme 8.2 Daten anordnen mit Tabellen Beispiel: Buchhandlung Redundanzen, Anomalien und Inkonsistenzen Projekteinstieg: Tabellen 8.3 Daten filtern mit SQL 8.4 Komplexe Filter Aufgaben
2. Modellierung von relationalen Datenbanken a) Datenbankentwurf durch ER-Diagramme - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung einer Datenbankmodellierung b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf		8.5 Datenbankentwurf *Beispiel: Online-Buchhandel* *Datenanalyse und Entwurf* *8.6 Umsetzung des ER-Modells* *Entitätsmengen* *m:n-Beziehungen* *1:n-Beziehungen* *1:1-Beziehungen* *Wiederaufgriff des Projekteinstiegs* *8.7 Datenbanken verbessern*

- Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln c) Normalformen - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)		*durch Normalformen*
3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken		Prüfungsvorbereitung

Hinweise:
- Die Texte sind aus dem neuen Lehrwerk Informatik 2 des Schöningh Verlags und dessen Vorwort bzw. dessen Kommentierungen entlehnt.
- Das vorliegende Übersichtsraster und die Sequenzierungen sollen am Ende des Schuljahrs 2015/2016 für die Q1 und am Ende des Schuljahrs 2016/2017 für die Q2 evaluiert und überarbeitet werden.

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Goerdeler-Gymnasiums die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
4) Medien und Arbeitsmittel sind schülernah gewählt.
5) Die Schüler/innen erreichen einen Lernzuwachs.
6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
9) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
14) Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

15) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
16) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
17) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
18) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
19) Der Unterricht ist projekt- und produktorientiert angelegt.
20) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
21) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOSt sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Goerdeler-Gymnasiums im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:
Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr
  Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr
  Dauer der Klausuren: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren
  Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

Kriterien

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt auf der Basis einer Musterlösung mit Hilfspunkten, aus deren Formulierung zusammen mit den Klausurkorrektur-Anmerkungen der Erwartungshorizont und die Bewertungskriterien hervorgehen.

Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsumme zu den Notenstufen an dem Zuordnungsschema des Zentralabiturs.
Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOSt §13 (2)) angemessen erscheint.

Die Note ausreichend (5 Punkte) soll bei Erreichen von 45 % der Hilfspunkte erteilt werden.

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn des Schuljahres genannt.

Leistungsaspekte

Mündliche Leistungen
- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen

Praktische Leistungen am Computer
- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
  In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, können maximal zwei schriftliche Übungen pro Kurshalbjahr stattfinden. Über die Durchführung entscheidet die Lehrkraft.
  Schriftliche Übungen dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

Die folgenden allgemeinen Kriterien gelten sowohl für die mündlichen als auch für die schriftlichen Formen der sonstigen Mitarbeit.

Die Bewertungskriterien stützen sich auf
- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei zu legen auf
- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion.

Bei Gruppenarbeiten auch auf
- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und

- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf
- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen
- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann
- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

Vorbereitung auf die Erstellung der Facharbeit
Möglichst schon zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

Teilnahme an Wettbewerben
Den Schülerinnen und Schülern wird entsprechend ihres Alters regelmäßig die Teilnahme an Informatikwettbewerben empfohlen. Dazu werden die Wettbewerbe und einige Aufgaben vorgestellt. Dies umfasst neben dem Bundeswettbewerb Informatik auch praxisorientierte

Wettbewerbe, z.B. im Roboterbau (First Lego League, ZDI Roboterwettbewerb, Robocup etc.).

4 Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.

Das schulinterne Curriculum (siehe 2.1) wurde 2014 zum ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Es wird im Rahmen von Dienstbesprechungen regelmäßig evaluiert und mögliche Optimierungen der Fachkonferenz vorgestellt, in ihr diskutiert und beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.