

1 Rahmenbedingungen der fachlichen Arbeit

Das Mercator-Gymnasium ist eines von drei Innenstadtgymnasien Duisburgs. Seiner Lage zwischen dem Dellviertel und Hochfeld entsprechend besuchen Schülerinnen und Schüler unterschiedlicher Nationalitäten und Muttersprachen die Schule.

Die gymnasiale Oberstufe mit etwa 400-450 Schülerinnen und Schüler ist relativ groß, wodurch insgesamt gute Kurswahlmöglichkeiten gegeben sind. Neben den hauseigenen Schüler/innen aus der SI werden jährlich ca. 30 % neue Schülerinnen vor allem aus der Realschule aufgenommen.

Das Fach Informatik wird am Mercator-Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) zweistündig unterrichtet und von etwa der Hälfte der Schülerinnen und Schüler besucht. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Algorithmik am Beispiel einer didaktischen Lernumgebung und auf die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen eingegangen. Der Umgang mit informatischen Systemen und der sichere Umgang mit den neuen Medien haben hier einen großen Stellenwert.

In der Sekundarstufe II bietet das Mercator-Gymnasium für die eigenen Schülerinnen und Schüler in allen Jahrgangsstufen jeweils Grundkurse und Leistungskurse (Kooperation) 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 Programmierumgebung zum Einsatz, welche das Erlernen 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.

Mit momentan vier Informatiklehrer/innen ist die Schule sehr gut ausgestattet, so dass die Stundentafel im Fach Informatik und im WP II Bereich in Mathematik/ Informatik erfüllt werden kann.

Die Nähe zu den benachbarten Innenstadtgymnasien und Gesamtschulen ermöglicht im Rahmen der Kooperation die Einrichtung von Leistungskursen.

Die Schule verfügt über zwei große, verdunkelbare Informatikräume. Beide Räume befinden sich in Ebene 2 und sind benachbart. Raum 206 stellt für die Schülerinnen und Schüler 24

Computerarbeitsplätze zur Verfügung. Zusätzlich gibt es einen Lehrer PC. In diesem Raum befindet sich ein interaktiver Beamer. Im Raum 207 befinden sich 30 Computerarbeitsplätze und ein Lehrer PC. Hier gibt es ein interaktives Smartboard und zwei Whiteboards. Beide Rechnerräume teilen sich einen Drucker. Alle Arbeitsplätze sind an das schulinterne Rechnernetz (I-Serv) angeschlossen, so dass Schülerinnen und Schüler über einen individuell gestaltbaren Zugang zum zentralen Server der Schule alle Arbeitsplätze der zwei Räume zum Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden können.

 Einführungsphase (EF) - Grobübersichtsraster: Unterrichtsvorhaben

Nr.	Beschreibung
1	Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten
2	Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von Beispielkontexten
3	Thema: Grundlagen der objektorientiert Programmierung und algorithmischer Grundstrukturen in Java
4	Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von einem Sortieralgorithmus
5	Thema: Geschichte der digitalen Datenverarbeitung und Grundlagen des Datenschutzes

Unterrichtsvorhaben – Qualifikationsphase 1 (Q1)

Nr.	Beschreibung
1	Thema: Endliche Automaten und formale Sprachen
2	Thema: Verschlüsselung
3	Thema: Modellierung und Implementation von Anwendungen mit dynamischen, linearen Datenstrukturen
4	Thema: Suchen und Sortieren auf linearen Datenstrukturen

Unterrichtsvorhaben – Qualifikationsphase 2 (Q2)

Nr.	Beschreibung
1	Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten
2	Thema: Graphentheorie
3	Thema: Projektarbeit
4	Thema: Datenschutz



Einführungsphase - Übersichtsraster Unterrichtsvorhaben

Einführungsphase	
<u>Unterrichtsvorhaben EF I</u> Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten Zentrale Kompetenzen: • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatiksysteme • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Einzelrechner • Dateisystem • Intranet und Internet • Einsatz von Informatiksystemen Zeitbedarf: 6 Stunden	<u>Unterrichtsvorhaben EF II</u> Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von Beispielkontexten Zentrale Kompetenzen: • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen (und Automaten) Inhaltliche Schwerpunkte: • Objekte und Klassen • Syntax und Semantik einer Programmiersprache Zeitbedarf: 10 Stunden

Einführungsphase	
<u>Unterrichtsvorhaben EF III</u> Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Kommunizieren und kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen (und Automaten) Inhaltliche Schwerpunkte: • Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen Zeitbedarf: 24 Stunden	<u>Unterrichtsvorhaben EF IV</u> Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von einem Sortieralgorithmus Zentrale Kompetenzen: • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und kooperieren Inhaltsfelder: • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementation einfacher Algorithmen Zeitbedarf: 22 Stunden

Einführungsphase	
<u>Unterrichtsvorhaben EF V</u> Thema: Geschichte der digitalen Datenverarbeitung und Grundlagen des Datenschutzes Zentrale Kompetenzen: • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren Inhaltsfelder: • Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Der gläserne Deutsche • Automatisierungstechnik • Datenschutz Zeitbedarf: 12 Stunden	

2.2 Einführungsphase - Konkretisierte Unterrichtsvorhaben

Eingeführtes Lehrwerk Informatik I

Unterrichtsvorhaben EF I

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

Leitfragen: Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandenen informatische Ausstattung genutzt werden? Welches Vorwissen habe die Schüler?

Vorhabenbezogene Konkretisierung

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Zunächst wird der Informatikunterricht klar von einem „Programmierkurs“ abgegrenzt. Beginnend mit einem reduzierten Rechnermodell (Papier und Streichholzrechner) werden die Vorkenntnisse der SuS nivelliert. Nach der Thematisierung des Begriffes „Information“ wird auf die Übertragung von Informationen bzw. Daten im Sinne des „Sender-Empfänger-Modells“ eingegangen. Dabei wird eine überblickartige Vorstellung der Kommunikation von Rechnern in Netzwerken erarbeitet. Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der „Von-Neumann-Architektur“ erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden. Durch den konkreten Bezug zur informatischen Ausstattung der Schule wird in die verantwortungsvolle Nutzung eingeführt.

Zeitbedarf: 6 Stunden

Lernmittel/ Material

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Informationen, deren Kodierung und Speicherung (a) Informatik als Wissenschaft der Verarbeitung von Informationen (b) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner (c) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schulrechnern	Die SuS... • nutzen die im Unterricht eingesetzten Informatiksystem selbständig, sicher, zielführend und verantwortungsbewusst (D).	Papier und Streichholzrechner
2. Informations- und Datenübermittlung in Netzen (a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation (b) Informatische Kommunikation in Rechnernetzen an einem Beispiel (d) Richtlinien zum verantwortungsbewussten Umgang mit dem Internet	• nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).	Rollenspiel zur Paketvermittlung im Internet Informatik 3 : Kapitel 1
3. Aufbau informatischer Systeme	• beschreiben und erläutern den Aufbau und die	

(a) Herleitung der „Von-Neumann-Architektur“ (b) Identifikation des EVA-Prinzip als Prinzip der Verarbeitung von Daten als Grundlage der „Von-Neumann-Architektur“	Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A)	
---	---	--

Unterrichtsvorhaben EF II

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

Leitfragen: Was bedeutet Objektorientierung? Wie setzte ich das in Greenfoot um?

Vorhabenbezogene Konkretisierung

Zunächst wird auf den Begriff der objektorientierten Programmierung eingegangen. Ausgehend von der abstrakten Form bzw. dem Stoff (Zitat Gaarder) wird der Unterschied zwischen Klasse und Objekt entwickelt. Anschließend wird die Spezialisierung einer Oberklasse durch eine Unterklasse als wichtigstes Konzept der OOP vorgestellt. Anwendungen im Kontext ermöglichen den SuS einen Bezug zur Realität herzustellen.

Des Weiteren wird die Klasse Roboter mit Attributen und Methoden eingeführt und ein grundlegendes Verständnis von einfachen Algorithmen angebahnt. Die Implementation von einfachen Methoden in Greenfoot schließt sich logisch an.

Zeitbedarf : 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Die objektorientierte Modellierung (a) Anwendungsfälle (b) Vererbung über Spezialisierung und Darstellung im Klassendiagramm	Die SuS... • erläutern den Unterschied zwischen Klasse und Objekt (K) • erarbeiten die Vererbung über die Spezialisierung (D) • lösen Aufgaben im Anwendungskontext (M)	Kapitel 1 z.B. Kapitel 1.6 Training
2. Die Modellierung in Greenfoot (a) Methoden (b) Attribute, Aktionen, Vererbung (C) Objektidentität und Objektzustand	• analysieren die Klasse Roboter mit Attribute und Methoden (A) • bewerten einfache Algorithmen(D)	u.a. Seite 24, Aufgabe 5
3. Die Implementierung in Greenfoot	• übertragen den Entwurf in das Editorfenster (D) • benutzen erste Methoden (I)	u.a. S.25-26 u.a Seite 27, Aufgaben 3- 5

Unterrichtsvorhaben EF III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java

Leitfragen: Wie kommt Flexibilität in die Programmierung? Wie erreiche ich es, dass der Roboter nicht vor die Wand läuft bzw. das tut, was ich möchte?

Vorhabenbezogene Konkretisierung

Die Möglichkeit, Entscheidungen zu treffen und diese zu implementieren leitet die erste bedingte Anweisung ein. Die SuS analysieren verschiedene Darstellungsarten eines Algorithmus. Die anschließende Verknüpfung der Robotersensoren (UND, ODER, NICHT) erweitert das Verhalten der Roboter. Der Roboter kann nun selbständig nach vorher festgelegten Regeln auf verschiedene Situationen reagieren. Nun werden Schleifen eingeführt, modelliert und implementiert. Abschließend mündet alles in der selbständigen Implementierung von Algorithmen durch die Schüler.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Bedingte Anweisungen (a) Die If Anweisung (b) Pseudocode, PAP, Stylekonventionen	Die SuS... - modellieren die Bedingte Anweisung (M) - analysieren die Darstellung in Java, als PAP und in Pseudocode (A)	Kapitel 2 u.a. Seite 36-41
2. Logische Verknüpfungen (a) Und, Oder, Nicht und Kombinationen	implementieren logische Verknüpfungen (I)	u.a. Seite 41-47
3. Die Vor- und Nachprüfende Wiederholung	implementieren Wiederholungen (I)	u.a. Seite 48-53
4. Die Zählschleife	implementieren Zählschleifen (I)	u.a. Seite 53-57
5. Algorithmen entwickeln Geschichte der Datenverarbeitung	- verwenden die Grundkonstrukte zur Algorithmenerstellung (M) - vergleichen den Entwurf von Algorithmen damals und heute (K) - implementieren eigene Algorithmen (I)	u.a. Seite 57-64 Training Seiten 66 – 68 Paper Software-Engeneering (19988/89)

Unterrichtsvorhaben EF IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von einem Sortieralgorithmus

Leitfragen: Wie kann ich Verantwortlichkeiten eines Objekts und/oder Teilprobleme einer Lösung bündeln? Wie schreibe ich selbst eine Klasse?

Vorhabenbezogene Konkretisierung

Ausgehend von dem bereits benutzten Roboterszenario implementieren sie SuS eigene Methoden und ihre Dokumentationen. Sie erweitern ihre Kenntnisse über Variablen und Rückgabewerte und implementieren sie kontextbezogen. Eigene Klassen werden erst modelliert und dann implementiert. Hierbei werden die Assoziation (Kennt-Beziehung), die Objektexemplierung und die Datenkapselung eingeführt. Abschließend werden die Kenntnisse in einem Miniprojekt zusammenfassend vertieft.

Lernmittel/ Material		
Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1.Methoden, Variablen (a) Methodendefinition, Methodenaufruf, Methodendokumentation (b) Attribute definieren, Typen von Variablen, Attribute Werte zuweisen (c) Sortieren durch direkte Auswahl	Die SuS - implementieren Methoden (I) - implementieren Variablen und Attribute (I) - implementieren SelectionSort (I)	Informatik 1 u.a. S. 74-77 u.a. S.77-85 Informatik 2 u.a. S.114-116
2.Rückgaben und Parameter (a) Methoden mit Rückgaben (b) Methoden mit Parametern (c) Konstruktor	- implementieren Methoden mit Rückgaben (I) - implementieren Methoden mit Parametern (I) - implementieren den Konstruktor (I)	Informatik 1 u.a. Seite 85-87 u.a. Seiten 87-88 u.a. Seiten 88-91
3. Klassenentwurf (a) Modellierung der Klasse (b) Beziehung zu anderen Klassen (c) Implementierung der Klasse	- modellieren eine Klasse und deren Beziehungen (M) - implementieren eine Klasse (I)	Informatik 1 u.a. S. 100-104
4. Objekte (a) Objektexemplierung (b) Datenkapselung	- modellieren Objekte (M) - implementieren Datenkapselung einschließlich get- und set-Methoden (I)	Informatik 1 u.a. S. 104-107 u.a. S. 107-110
5.Projekte	Implementieren ein Minispiel	Informatik 1 z.B. Tortenschlacht, Greeps, Hase-Fuchs Szenario u.a. S.111-113

Unterrichtsvorhaben EF V

Thema: Geschichte der digitalen Datenverarbeitung und Grundlagen des Datenschutzes

Leitfragen: Wie verändert die Informatik die Gesellschaft? Was bedeutet das für den Einzelnen?

Vorhabenbezogene Konkretisierung

Ausgehend von einer aktuellen Dokumentation erarbeiten die SuS arbeitsteilig verschiedenen Aspekte des Datenschutzes (Erfassung und Verarbeitung persönlicher Daten, Observierungen und staatliche Mittel zur Überwachung) . Datenschutz als Grundrecht wird mit Datenschutz im WWW abgeglichen und kritisch beleuchtet. Im zweiten Teil der Reihe wird die Automatisierungstechnik betrachtet. Aktuelle Artikel werden analysiert, interpretiert und diskutiert.

Lernmittel/ Material

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Der gläserne Deutsche	Die SuS... - interpretieren eine Dokumentation zum Thema Datenschutz(D) - kommunizieren Teilaspekte des Themas (K) - erarbeiten eine Präsentation zur Darstellung ihrer Ergebnisse(D)	Informatik 3, Kapitel 2 S.54-55 z.B. TV Dokumentation: „Der gläserne Deutsche – Wie wir Bürger ausgespäht werden“ Online Artikel zum Thema (/home/Informatik/Material)
2. Der gläserne Mensch (a) Der gläserne Mensch (b) Datenschutz als Grundrecht (C) Datenschutz im WWW	- argumentieren über Teilthemen(A) - stellen Teilthemen dar und interpretieren sie (D) - beziehen persönlich Stellung zum Thema (K)	u.a. Seite 54-65
3. Informatik und Ethik	Die SuS... - setzen sich mit den illegalen Möglichkeiten der heutigen Informatiksysteme auseinander (A) - beziehen persönlich Stellung zum Thema (K)	u.a. Informatik 3 S.58-65
4. Mensch und Technik- Wer hat die Oberhand? (a) Mensch gegen Maschine (b) Automatisierung im Alltag	- Argumentieren Teilthemen(A) - stellen Teilthemen dar und interpretieren sie (D) - beziehen persönlich Stellung zum Thema (K)	u. A. Seiten 75-81 z.B. Film: Robot und Frank

 Grobübersichtsraster: Unterrichtsvorhaben

Nr.	Beschreibung Qualifikationsphase 1 (Q1 GK)
1	Thema: Endliche Automaten und formale Sprachen
2	Thema: Wiederholung und Vertiefung der objektorientierten Modellierung
3	Thema: Modellierung und Implementation von Anwendungen mit dynamischen, linearen Datenstrukturen
4	Thema: Suchen und Sortieren auf linearen Datenstrukturen

Qualifikationsphase 2 (Q2)

Nr.	Beschreibung Qualifikationsphase 2 (Q2 GK)
1	Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten
2	Thema: Graphentheorie
3	Thema: Aufbau und Kommunikation in Netzwerken

3.2 Qualifikationsphase **Grundkurs** - Unterrichtsvorhaben

Qualifikationsphase1 (Q1 GK)	
<u>Unterrichtsvorhaben Q1 I</u> Thema: Endliche Automaten und formale Sprachen Zentrale Kompetenzen: • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und kooperieren Inhaltsfelder: • Endliche Automaten und formale Sprachen Inhaltliche Schwerpunkte: • Endliche Automaten • Grammatik regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen Zeitbedarf: 20 Stunden	<u>Unterrichtsvorhaben Q1 II</u> Thema: Wiederholung und Vertiefung der objektorientierten Modellierung Zentrale Kompetenzen: - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren Inhaltsfelder: - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft - Informatiksysteme Inhaltliche Schwerpunkte: - Objekte und Klassen - Wirkung der Automatisierung - Nutzung von Informatiksystemen Zeitbedarf: 14 Stunden

Qualifikationsphase1 (Q1 GK)	
<u>Unterrichtsvorhaben Q1 III</u>	<u>Unterrichtsvorhaben Q1 IV</u>

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

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementation von Algorithmen
- Algorithmen in ausgewählten Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 20 Stunden

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 16 Stunden

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen
- Syntax und Semantik einer Programmiersprache
- Sicherheit

Zeitbedarf: 20 Stunden

Thema: Graphentheorie

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

-

Zeitbedarf: 22 Stunden

Thema: Aufbau von und Kommunikation in Netzwerken

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen, Wirkungen der Automatisierung
- Grundprinzipien des Datenschutzes
 - Verbot mit Erlaubnisvorbehalt
 - Erforderlichkeit

Zeitbedarf: 16 Stunden

Unterrichtsvorhaben Q1 I

Thema: Endliche Automaten und formale Sprachen

Leitfragen: Wie lassen sie Automaten beschreiben? Welcher Zusammenhang besteht zwischen einem Automaten und einer (regulären) Grammatik? Kann ein NEA mehr als ein DEA?

Vorhabenbezogene Konkretisierung

Ausgehend von einer Mausefalle als kleinsten Automaten erarbeiten sich die SuS den Begriff des deterministischen endlichen Automaten(DEA). Die Definition und Darstellung als Graph und Tabelle wird eingeführt. Zahlreiche Übungen sowohl von Automat zur Aufgabe des Automaten als auch vice versa vertiefen das Verständnis. Die Erweiterung zum Nichtdeterministischen endlichen Automaten(NEA) führt zur Kernfrage der Automatentheorie: Kann ein NEA mehr als ein DEA? Hier spielt bereits der Begriff der Grammatik eine Rolle. Reguläre und kontextfreie Sprachen schließen hier logisch an. Der Parser als erkennender Automat rundet die Sequenz ab. Als nächstes erarbeiten und präsentieren die SuS arbeitsteilig Mealy/Moore/ Turing Automaten, Kurzbiografien zu Alan Turing und Noam Chomsky. Abschließend wird eine vertiefte Darstellung mit Hilfe der Chomsky Hierarchie erreicht.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. DEA 2. NEA 3. Transformation NEA in DEA 4. Mealy- Automat 5. Turing-Maschine 6. KellerAutomat	Die SuS... • modellieren Automaten (M) • vergleichen Automaten (A) • stellen Automaten dar(D) • kommunizieren und kooperieren in arbeitsteiligen Gruppen(K)	u.a. Klett: Informatik 3, Schöningh: Informatik 3,
1. Formale Sprachen und Grammatik (a) reguläre Sprachen (b) kontextfreie Sprachen (c) Parser	• modellieren Sprachen (M) • vergleichen Sprachen (A) • stellen Sprachen dar(D)	Auch: Produktionen mit ε

Zeitbedarf: 20 Stunden

Unterrichtsvorhaben Q1 II

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

Leitfragen: Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

Vorhabenbezogene Konkretisierung:

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

Zeitbedarf: 14 Stunden

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 ... - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - analysieren und erläutern objektorientierte	Kapitel 1 Konzepte des objektorientierten Modellierens 1.1 Modellierung der Realität 1.2 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

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),
- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie
- 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).

1.3 Gut geplant – Klassenentwurf

1.4 Hierarchien machen's einfacher – Vererbung

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

Leitfragen:

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

Vorhabensbezogene Konkretisierung:

Ausgehend von Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der Stack als auch die Queue benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: 20 Stunden

Vorhabenbezogene Konkretisierung

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien und Materialien
1. Die Datenstruktur Feld a) Erarbeitung der Anforderungen an eine Datenstruktur b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays c) Modellierung und Implementierung von Anwendungen	Die Schülerinnen und Schüler ... - erläutern Operationen dynamischer (linearer) Datenstrukturen (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),	Kapitel 2 Lineare Datenstrukturen 2.1 Anforderungen an eine Datenstruktur 2.2 Datenansammlungen fester Größe – Arrays
2. Die Datenstruktur Schlange a) Modellierung und Implementierung der Verknüpfung von Objekten b) Generische Typen, Trennung von	- interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D)	2.3 Wer zuerst kommt ... – Schlangen

Verwaltung und Inhalt dyn. DS. c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i>	- 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) - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M).	
3. Die Datenstruktur Stapel a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i>, Erschließen der Standardoperationen c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i> d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array</i> (<i>Palindrom</i>))		2.4 Daten gut abgelegt – Stapel
4. Die Datenstruktur Liste a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>) b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i>		2.5 Flexibel für alle Fälle – lineare Listen

- | | | |
|--|--|--|
| c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i>
d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List) | | |
|--|--|--|

Unterrichtsvorhaben Q1 IV**Thema:** Suchen und Sortieren auf linearen Datenstrukturen**Leitfragen:** Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?**Vorhabenbezogene Konkretisierung:**

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmusentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

Zeitbedarf: 20 Stunden**Vorhabenbezogene Konkretisierung**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Eigenschaften von Algorithmen a) Qualitätseigenschaften von Algorithmen b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“ c) Analyse und Entwicklung von rekursiven Algorithmen	Die Schülerinnen und Schüler ... - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),	Kapitel 3 Algorithmen 3.1 Ohne Algorithmen läuft nichts 3.2 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

- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),
- 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),
- 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),.

3.3 Suchen – iterativ und rekursiv
 Projekteinstieg: Laufzeitanalyse
 experimentell

3. Sortieren auf Listen und Arrays

- a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen)
- b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)
- c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf
- d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)

3.4 Sortieren – iterativ und rekursiv

Unterrichtsvorhaben Q2 I**Thema:** Modellierung und Nutzung von Datenbanken in Anwendungskontexten**Leitfragen:** Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?**Vorhabenbezogene Konkretisierung**

Ausgehend von überschaubaren Datenbanken entwickeln SuS relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Die notwendigen Grundbegriffe und Fertigkeiten sowie die erforderlichen einfachen SQL-Abfragen werden hier erarbeitet.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden. Dafür entwickeln die SuS in den Anwendungssituationen Entitäten, Attribute, Relationen und Kardinalitäten und stellen sie graphisch in Entity-Relationship Diagrammen dar. Die Diagramme werden interpretiert, erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. An Beispielen wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Zeitbedarf: 20 Stunden

Lernmittel/ Material

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Nutzung von relationalen Datenbanken (a) Aufbau von Datenbanken und Grundbegriffen - Entwicklung von Fragestellungen zur vorhandenen Datenbank - Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema (b) SQL-Abfragen - Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT, DISTINCT, AND, OR, NOT) auf einer Tabelle - Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabellen zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren, Vergleichsoperatoren) (c) Vertiefung an weiteren Beispielen	Die SuS... - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A). - bestimmen Primär- und Sekundärschlüssel (M). - verwenden die Fachsprache bei der Kommunikation über fachliche Inhalte (K). - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D). - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A). - verwenden die Syntax und Semantik einer Datenbankabfragesprache um Informationen aus einem Datenbanksystem zu extrahieren (I).	Klett: Informatik2 , Schöningh: Informatik3

2. Modellierung von relationalen Datenbanken**(a) Entity-Relationship Diagramm**

- Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship Diagramms
- Erläuterung einer Datenbankmodellierung
- Modifizierung einer Datenbankmodellierung

(b) Entwicklung einer Datenbank aus einem Datenbankentwurf

- Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship Diagramm

- Bestimmung von Primär- und Sekundärschlüsseln

(c) Redundanz, Konsistenz, Normalformen

- Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation
- Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung

Die SuS...

- ermitteln für anwendungsbezogene Problemstellungen, Entitäten, Attribute, Relationen und Kardinalitäten (M).
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship Diagramm graphisch dar (D).
- modifizieren eine Datenbankmodellierung (M).
- analysieren und erläutern eine Datenbankmodellierung (A)
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).
- analysieren und erläutern eine Datenbankmodellierung (A).
- modellieren zu einem Entity-Relationship Diagramm ein relationales Datenbankschema (M).
- bestimmen Primär- und Sekundärschlüssel (M).
- modifizieren eine Datenbankmodellierung (M).
- analysieren und erläutern eine Datenbankmodellierung (A).
- erläutern die Eigenschaften normalisierter Datenbankschemata (A).
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).
- überführen Datenbankschemata in vorgegebene Normalformen (M).

Unterrichtsvorhaben Q2 II

Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Leitfragen: Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

Vorhabenbezogene Konkretisierung:

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche). Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse *BinaryTree* (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse *BinaryTree* thematisiert.

Möchte man Daten geordnet speichern, bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert. Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces *ComparableContent* realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen *BinarySearchTree* und *ComparableContent* modelliert und (ggf. in Teilen) implementiert.

Zeitbedarf: 20 Stunden

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),	Kapitel 5 Nicht-lineare Datenstrukturen 5.1 Spielen mit Struktur – Baumstrukturen Projekteinstieg 1: Spielbäume
2. Binäre Bäume a) rekursiver Aufbau eines binären Baums b) Traversierungen (pre-, in-, postorder)		5.2 Zwei Nachfolger sind genug! - Binäre Bäume Implementation des Projekts Ahnenbaum

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)	- 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),	
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen	- 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).	

Unterrichtsvorhaben Q2-III

Thema: Aufbau von und Kommunikation in Netzwerken

Leitfragen: Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/ informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden?

Vorhabenbezogene Konkretisierung:

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

Zeitbedarf: 12 Stunden

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 Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), - nutzen bereitgestellte Informatiksysteme und das Internet	Kapitel 6 Kommunikation in Netzwerken 6.1 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	reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	6.2 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		6.3 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 111 – Kryptologie

Qualifikationsphase (Q1/Q2 LK)

Grobübersichtsraster: Unterrichtsvorhaben

Nr.	Beschreibung Qualifikationsphase 1 (Q1 LK)
1	Thema: Endliche Automaten und formale Sprachen
2	Thema: Wiederholung und Vertiefung der objektorientierten Modellierung
3	Thema: Modellierung und Implementation von Anwendungen mit dynamischen, linearen Datenstrukturen
4	Thema: Suchen und Sortieren auf linearen Datenstrukturen

Qualifikationsphase 2 (Q2)

Nr.	Beschreibung Qualifikationsphase 2 (Q2 GK)
1	Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten
2	Thema: Graphentheorie
3	Thema: Aufbau und Kommunikation in Netzwerken

4.2 Qualifikationsphase **Leistungskurs** - Unterrichtsvorhaben

Qualifikationsphase1 (Q1 LK)	
<u>Unterrichtsvorhaben Q1 I</u> Thema: Endliche Automaten und formale Sprachen Zentrale Kompetenzen: • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und kooperieren Inhaltsfelder: • Endliche Automaten und formale Sprachen Inhaltliche Schwerpunkte: • Endliche Automaten • Grammatik regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen Zeitbedarf: 35 Stunden	<u>Unterrichtsvorhaben Q1 II</u> Thema: Wiederholung und Vertiefung der objektorientierten Modellierung Zentrale Kompetenzen: - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren Inhaltsfelder: - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft - Informatiksysteme Inhaltliche Schwerpunkte: - Objekte und Klassen - Wirkung der Automatisierung - Nutzung von Informatiksystemen Zeitbedarf: 35 Stunden

Qualifikationsphase1 (Q1 LK)	
<u>Unterrichtsvorhaben Q1 III</u>	<u>Unterrichtsvorhaben Q1 IV</u>

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

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementation von Algorithmen
- Algorithmen in ausgewählten Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 35 Stunden

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

Zeitbedarf: 20 Stunden

Unterrichtsvorhaben Q2 I

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen
- Syntax und Semantik einer Programmiersprache
- Sicherheit

Zeitbedarf: 40 Stunden

Unterrichtsvorhaben Q2 II

Thema: Graphentheorie

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und kooperieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

-

Zeitbedarf: 30 Stunden

Unterrichtsvorhaben Q2 III

Thema: Aufbau von und Kommunikation in Netzwerken

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen, Wirkungen der Automatisierung
- Grundprinzipien des Datenschutzes
 - Verbot mit Erlaubnisvorbehalt
 - Erforderlichkeit

Zeitbedarf: 22 Stunden

Qualifikationsphase **LK** - Konkretisierte Unterrichtsvorhaben

Unterrichtsvorhaben Q1 I

Thema: Endliche Automaten und formale Sprachen

Leitfragen: Wie lassen sie Automaten beschreiben? Welcher Zusammenhang besteht zwischen einem Automaten und einer (regulären) Grammatik? Kann ein NEA mehr als ein DEA?

Vorhabenbezogene Konkretisierung

Ausgehend von einer Mausefalle als kleinsten Automaten erarbeiten sich die SuS den Begriff des deterministischen endlichen Automaten(DEA). Die Definition und Darstellung als Graph und Tabelle wird eingeführt. Zahlreiche Übungen sowohl von Automat zur Aufgabe des Automaten als auch vice versa vertiefen das Verständnis. Die Erweiterung zum Nichtdeterministischen endlichen Automaten(NEA) führt zur Kernfrage der Automatentheorie: Kann ein NEA mehr als ein DEA? Hier spielt bereits der Begriff der Grammatik eine Rolle. Reguläre und kontextfreie Sprachen schließen hier logisch an. Der Parser als erkennender Automat rundet die Sequenz ab. Als nächstes erarbeiten und präsentieren die SuS arbeitsteilig Mealy/Moore/ Turing Automaten, Kurzbiografien zu Alan Turing und Noam Chomsky. Abschließend wird eine vertiefte Darstellung mit Hilfe der Chomsky Hierarchie erreicht.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. DEA 2.NEA 3.Transformation NEA in DEA 4. Mealy- Automat 5. Turing-Maschine 6. Kellerautomat	Die SuS... - modellieren Automaten (M) - vergleichen Automaten (A) - stellen Automaten dar(D) - kommunizieren und kooperieren in arbeitsteiligen Gruppen(K)	u.a. Klett: Informatik 3, Schöningh: Informatik 3,
1. Formale Sprachen und Grammatik (a) reguläre Sprachen (b) kontextfreie Sprachen (C) Parser	- modellieren Sprachen (M) - vergleichen Sprachen (A) - stellen Sprachen dar(D)	Auch: Produktionen mit ϵ

Zeitbedarf: 35 Stunden

Unterrichtsvorhaben Q1 II

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

Leitfragen: Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

Vorhabenbezogene Konkretisierung:

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

Zeitbedarf: 35 Stunden

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 ... - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - analysieren und erläutern objektorientierte	Kapitel 1 Konzepte des objektorientierten Modellierens 1.1 Modellierung der Realität 1.2 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

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),
- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie
- 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).

1.3 Gut geplant – Klassenentwurf

1.4 Hierarchien machen's einfacher – Vererbung

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

Leitfragen:

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

Vorhabensbezogene Konkretisierung:

Ausgehend von Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der Stack als auch die Queue benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können.

Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: 35 Stunden

Vorhabenbezogene Konkretisierung

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien und Materialien
1. Die Datenstruktur Feld a) Erarbeitung der Anforderungen an eine Datenstruktur b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays c) Modellierung und Implementierung von Anwendungen	Die Schülerinnen und Schüler ... - erläutern Operationen dynamischer (linearer) Datenstrukturen (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),	Kapitel 2 Lineare Datenstrukturen 2.1 Anforderungen an eine Datenstruktur 2.2 Datenansammlungen fester Größe – Arrays
2. Die Datenstruktur Schlange a) Modellierung und Implementierung der Verknüpfung von Objekten b) Generische Typen, Trennung von	- interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D)	2.3 Wer zuerst kommt ... – Schlangen

Verwaltung und Inhalt dyn. DS. c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i>	- 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) - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M).	
3. Die Datenstruktur Stapel a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i>, Erschließen der Standardoperationen c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i> d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array</i> (<i>Palindrom</i>))		2.4 Daten gut abgelegt – Stapel
4. Die Datenstruktur Liste a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>) b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i>		2.5 Flexibel für alle Fälle – lineare Listen

- | | | |
|--|--|--|
| c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i>
d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List) | | |
|--|--|--|

Unterrichtsvorhaben Q1 IV**Thema:** Suchen und Sortieren auf linearen Datenstrukturen**Leitfragen:** Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?**Vorhabenbezogene Konkretisierung:**

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmusentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

Zeitbedarf: 20 Stunden**Vorhabenbezogene Konkretisierung**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Eigenschaften von Algorithmen a) Qualitätseigenschaften von Algorithmen b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“ c) Analyse und Entwicklung von rekursiven Algorithmen	Die Schülerinnen und Schüler ... - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),	Kapitel 3 Algorithmen 3.1 Ohne Algorithmen läuft nichts 3.2 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

- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),
- 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),
- 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),.

3.3 Suchen – iterativ und rekursiv
 Projekteinstieg: Laufzeitanalyse experimentell

3. Sortieren auf Listen und Arrays

- a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen)
- b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)
- c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf
- d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)

3.4 Sortieren – iterativ und rekursiv

Unterrichtsvorhaben Q2 I

Thema: Modellierung und Nutzung von Datenbanken in Anwendungskontexten

Leitfragen: Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

Vorhabenbezogene Konkretisierung

Ausgehend von überschaubaren Datenbanken entwickeln SuS relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Die notwendigen Grundbegriffe und Fertigkeiten sowie die erforderlichen einfachen SQL-Abfragen werden hier erarbeitet.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden. Dafür entwickeln die SuS in den Anwendungssituationen Entitäten, Attribute, Relationen und Kardinalitäten und stellen sie graphisch in Entity-Relationship Diagrammen dar. Die Diagramme werden interpretiert, erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. An Beispielen wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Zeitbedarf: 40 Stunden

Lernmittel/ Material

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
1. Nutzung von relationalen Datenbanken (a) Aufbau von Datenbanken und Grundbegriffen - Entwicklung von Fragestellungen zur vorhandenen Datenbank - Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema (b) SQL-Abfragen - Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT, DISTINCT, AND, OR, NOT) auf einer Tabelle - Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabellen zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren, Vergleichsoperatoren) - Vertiefung an weiteren	Die SuS... - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A). - bestimmen Primär- und Sekundärschlüssel (M). - verwenden die Fachsprache bei der Kommunikation über fachliche Inhalte (K). - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D). - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A). - verwenden die Syntax und Semantik einer Datenbankabfragesprache um Informationen aus einem Datenbanksystem zu extrahieren (I).	Klett: Informatik2 , Schöningh: Informatik3

Beispielen		
2. Modellierung von relationalen Datenbanken (a) Entity-Relationship Diagramm • Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship Diagramms • Erläuterung einer Datenbankmodellierung • Modifizierung einer Datenbankmodellierung (b) Entwicklung einer Datenbank aus einem Datenbankentwurf • Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship Diagramm • Bestimmung von Primär- und Sekundärschlüsseln (c) Redundanz, Konsistenz, Normalformen • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung	Die SuS... • ermitteln für anwendungsbezogene Problemstellungen, Entitäten, Attribute, Relationen und Kardinalitäten (M). • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship Diagramm graphisch dar (D). • modifizieren eine Datenbankmodellierung (M). • analysieren und erläutern eine Datenbankmodellierung (A) • präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K). • analysieren und erläutern eine Datenbankmodellierung (A). • modellieren zu einem Entity-Relationship Diagramm ein relationales Datenbankschema (M). • bestimmen Primär- und Sekundärschlüssel (M). • modifizieren eine Datenbankmodellierung (M). • analysieren und erläutern eine Datenbankmodellierung (A). • erläutern die Eigenschaften normalisierter Datenbankschemata (A). • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). • überführen Datenbankschemata in vorgegebene Normalformen (M).	

Unterrichtsvorhaben Q2 II

Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Leitfragen: Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

Vorhabenbezogene Konkretisierung:

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse *BinaryTree* (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse *BinaryTree* thematisiert.

Möchte man Daten geordnet speichern, bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Lösches und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces *ComparableContent* realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen *BinarySearchTree* und *ComparableContent* modelliert und (ggf. in Teilen) implementiert.

Zeitbedarf: 30 Stunden

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).	Kapitel 5 Nicht-lineare Datenstrukturen 5.1 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 <i>BinaryTree</i> (als Entwurfs- und Implementationsdiagramm) d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)	- 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),	5.2 Zwei Nachfolger sind genug! - Binäre Bäume Implementation des Projekts Ahnenbaum

4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen

- 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).

Unterrichtsvorhaben Q2-III

Thema: Aufbau von und Kommunikation in Netzwerken

Leitfragen: Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/ informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden?

Vorhabenbezogene Konkretisierung:

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

Zeitbedarf: 22 Stunden

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 Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	Kapitel 6 Kommunikation in Netzwerken 6.1 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		6.2 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

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

6.3 Einer für alle – Client-Server-Struktur

Die digitale Welt 111 – Kryptologie

4 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 Mercator Gymnasiums im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen.

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
- Leistungskurse Q 1: 2 Klausuren je Halbjahr
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren
Dauer der Klausuren: 3 Unterrichtsstunden
- Leistungskurse Q 2.1: 2 Klausuren
Dauer der Klausuren: 4 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Leistungskurse 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 über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

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

- ggf. Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.
Schriftliche Übung 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 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 zu legen.

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.

Leistungsrückmeldungen erfolgen auch in der Einführungsphase im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.