

Maximilian Kolbe Gymnasium Köln

Schulinterner Lehrplan Informatik

Ergänzung Qualifikationsphase

Stand: 7.6.2015, HIE

Inhaltsverzeichnis

Übersichtsraster Unterrichtsvorhaben: Qualifikationsphase 1.....	3
Konkretisierte Unterrichtsvorhaben Q1.....	4
Unterrichtsvorhaben Q1-I.....	5
Unterrichtsvorhaben Q1-II.....	7
Unterrichtsvorhaben Q1-III.....	9
Unterrichtsvorhaben Q1-IV.....	11
Übersichtsraster Unterrichtsvorhaben: Qualifikationsphase 2.....	13
Unterrichtsvorhaben Q2-I.....	14
Unterrichtsvorhaben Q2-II.....	16
Unterrichtsvorhaben Q2-III.....	18

Übersichtsraster Unterrichtsvorhaben: Qualifikationsphase 1

<p><i>Unterrichtsvorhaben Q1-I</i></p> <p>Thema: Wiederholung und Vertiefung der objektorientierten Modellierung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Wirkung der Automatisierung <p>Zeitbedarf: 10 Stunden</p>	<p>Unterrichtsvorhaben Q1-II</p> <p>Thema: Organisation und Verarbeitung von Daten I – Modellierung und Implementierung dynamischer linearer Datenstrukturen und deren Anwendungen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Nutzung von Informatiksystemen <p>Zeitbedarf: 20 Stunden</p>
<p><i>Unterrichtsvorhaben Q1-III</i></p> <p>Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 20 Stunden</p>	<p>Unterrichtsvorhaben Q1-IV</p> <p>Thema: Netzwerke</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Einzelrechner und Rechnernetzwerke - Sicherheit - Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: 10 Stunden</p>

Konkretisierte Unterrichtsvorhaben Q1

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

Ebenso bieten fast alle Unterrichtsvorhaben, in denen Programme implementiert werden, die Gelegenheit, die folgenden Kompetenzen zu erwerben bzw. zu vertiefen:

Schülerinnen und Schüler

- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),

Unterrichtsvorhaben Q1-I

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: 10 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> - 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),
<p>2. Erweiterung der objektorientierten Programmierung</p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> - 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),
<p>3. Mensch und Technik</p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> - 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),
<p>4. Übung und Vertiefung der OOM / OOP</p>	<ul style="list-style-type: none"> - 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).

Unterrichtsvorhaben Q1-II

Thema: Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen

Leitfragen: Wie können beliebig viele linear angeordnete Daten im Anwendungskontext verwaltet werden?

Zeitbedarf: 25 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Stapeln am Beispiel dargestellt und die Operationen der Klasse Stack anhand der Abiturklasse erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Dabei werden die Operationen der Datenstruktur Schlange thematisiert und die entsprechende Abiturklasse Queue verwendet.

Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse List gemäß der Abiturklasse eingeführt und in einem Anwendungskontext verwendet.

(Je nach thematisierten Anwendungskontexten ist eine andere Reihenfolge bei der Behandlung der oben genannten Datenstrukturen möglich und zulässig.)

In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Lernmittel / Materialien:

- Arbeitsblätter
- eine didaktische Entwicklungsumgebung (z. B. Java-Editor o. a.)
- Zeigermodell
- Modellprogramm zur Visualisierung dynamischer Listenstrukturen

Unterrichtssequenzen	zu entwickelnde Kompetenzen
<p>1. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <ul style="list-style-type: none"> • Ermittlung von Objekten, ihren Eigenschaften und Operationen • Erarbeitung der Funktionalität der Klasse Stack • Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Stack. 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • 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 Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A)
<p>2. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse Queue</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Erarbeitung der Funktionalität der Klasse Queue • Modellierung und Implementierung der Anwendung unter Verwendung der Klasse Queue. 	
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p> <ul style="list-style-type: none"> • Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen • Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List. 	

Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum 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 Algorithmentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

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 thematisiert. 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 sowie ihre Effizienz bewertet.

Zeitbedarf: 20 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen
<p>1. Eigenschaften von Algorithmen</p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - analysieren und erläutern Algorithmen und Programme (A), - 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),
<p>2. Suchen in Listen und Arrays</p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - 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),
<p>3. Sortieren auf Listen und Arrays</p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - 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),

Unterrichtsvorhaben Q1-IV

Thema: Netzwerke

Leitfragen: Wie werden Daten in Netzwerken übermittelt? Wie kann Kommunikation geschützt werden?

Zeitbedarf: 10 Stunden (Das Thema bietet Vertiefungen darüber hinaus an vielen Stellen an. Eine Erprobung des Unterrichtsvorhabens kann über Möglichkeiten hierzu Aufschluss geben und den zeitlichen Bedarf ggf. verändern.)

Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von einer Kommunikation zwischen zwei Kommunikationspartnern über eine einfache Leitung wird die Notwendigkeiten einer Datenübertragung erarbeitet. Die Schichten des TCP/IP-Schichtenmodells werden beispielgebunden erarbeitet (Basisbandübertragungsverfahren, Prüfverfahren, Vermittlungsschicht, Anwendungsprotokoll) und an einer Simulationssoftware getestet. Verschiedene Netzwerk-Topologien werden entwickelt und in Client-Server-Anwendungen simuliert. Über die Sicherheit von Netzwerkanwendungen wird das Augenmerk auf verschiedene symmetrische und asymmetrische kryptografische Verfahren gelenkt, welche analysiert und erläutert werden.

Lernmittel / Materialien:

- Arbeitsblätter zur Einführung in Netzwerke
- Simulationsprogramm "Filius"
- Arbeitsblätter und Skript zu "Filius"

Unterrichtssequenzen	zu entwickelnde Kompetenzen
1. Schichten des TCP/IP-Protokolls <ul style="list-style-type: none"> • Erarbeitung der Notwendigkeiten einer Netzwerkkommunikation • Erarbeitung der Schichten des TCP/IP-Protokolls: Ethernet-, Internet-, Transport- und Anwendungsschicht 	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),
2. Simulation von Netzwerken / Netzwerk-Topologien <ul style="list-style-type: none"> • Erarbeitung der Topologien: Peer-to-Peer, Sterntopologie, Baumtopologie, Vermaschtes Netz • Simulation von Client-Server-Anwendungen • Simulation von Protokollen der Anwendungsschicht (POP3, SMTP, etc.) 	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),
3. Analyse und Erläuterung kryptografischer Verfahren <ul style="list-style-type: none"> • Erläuterung symmetrischer Verfahren: monoalphabetisch: Cäsar, polyalphabetisch: Vigenère • Erläuterung asymmetrischer Verfahren: RSA, Diffie-Hellman • Analyse der Sicherheit verschiedener Verfahren und Auswirkungen auf den Datenschutz/Urheberrecht 	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A). • 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).

Übersichtsraster Unterrichtsvorhaben: Qualifikationsphase 2

<p><i>Unterrichtsvorhaben Q2-I</i></p> <p>Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Objekte und Klassen - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 20 Stunden</p>	<p>Unterrichtsvorhaben Q2-II</p> <p>Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Daten und ihre Strukturierung - Algorithmen - Formale Sprache und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Datenbanken - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Nutzung von Informatiksystemen - Sicherheit - Wirkung der Automatisierung <p>Zeitbedarf: 10 Stunden</p>
<p>Unterrichtsvorhaben Q2-III</p> <p>Thema: Endliche Automaten und Formale Sprachen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Syntax und Semantik einer Programmiersprache - Endliche Automaten - Grammatiken regulärer Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Formale Sprachen und Automaten - Informatiksysteme <p>Zeitbedarf: 10 Stunden</p>	

Unterrichtsvorhaben Q2-I

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume

Leitfragen: Wie können Daten im Anwendungskontext mit Hilfe binärer Baumstrukturen verwaltet werden? Wie kann dabei der rekursive Aufbau der Baumstruktur genutzt werden? Welche Vor- und Nachteile haben Suchbäume für die geordnete Verwaltung von Daten?

Zeitbedarf: 20 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Es werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse `BinaryTree` der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) thematisiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse `BinarySearchTree` der Vorgaben für das Zentralabitur weitere Klassen oder Methoden in diesem Kontext modelliert und implementiert. Die Suchbäume werden wie zuvor auch grafisch dargestellt.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderer Kontexten weiter geübt.

Lernmittel / Materialien:

- eine didaktische Entwicklungsumgebung (z. B. Java-Editor o. a.)
- Arbeitsblätter und Demonstrationsprogramme

Unterrichtssequenzen	zu entwickelnde Kompetenzen
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ul style="list-style-type: none"> • Grundlegende Begriffe • Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • 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), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare und nichtlineare Datensammlungen zu (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • analysieren und erläutern Algorithmen und Programme (A), • 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), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext • Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms • Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen • Implementierung der Anwendung oder von Teilen der Anwendung • Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf 	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms, • grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften • Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation • Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums 	
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>	

Unterrichtsvorhaben Q2-II

Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

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

Zeitbedarf: 25 Stunden

Absprachen zur vorhabenbezogenen Konkretisierung:

Ausgehend von einer konkreten Anwendungssituation entwickeln die Schülerinnen und Schüler Ideen zur Modellierung von Daten und erkennen die Vorzüge von Datenbanksystemen.

In weiteren Anwendungskontexten müssen Datenbanken entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in das Relationale Modell überführt.

An einem Beispiel 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.

Ausgehend von einer vorhandenen Datenbasis entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. Die Operationen der Relationenalgebra werden mit SQL-Abfragen simuliert.

Anhand von Fallbeispielen werden Probleme bei der Nutzung von Datenbanksystemen aufgezeigt und im Hinblick auf gesellschaftliche Auswirkungen diskutiert.

(Die exakte Abfolge der Teilvorhaben sowie die Verzahnung theoretischer und praktischer Anteile kann hierbei je nach Anwendungskontext flexibel angepasst werden.)

Lernmittel / Materialien:

- Arbeitsblätter
- SQL-Tutorial (z. B. die Seite SQL-Zoo, VideoCenter-Datenbank)

Unterrichtssequenzen	zu entwickelnde Kompetenzen
<p>1. Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Entity-Relationship-Diagramm: Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms Erläuterung und Modifizierung einer Datenbankmodellierung • Entwicklung einer Datenbank aus einem Datenbankentwurf: Modellierung eines relationalen Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln • Redundanz, Konsistenz und Normalformen: Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation. Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • bestimmen Primär- und Sekundärschlüssel (M), • analysieren und erläutern eine Datenbankmodellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), • überführen Datenbankschemata in die 1. bis 3. Normalform (M).
<p>2. Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> • Aufbau von Datenbanken und Grundbegriffe: Entwicklung von Fragestellungen zur vorhandenen Datenbank Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema • SQL-Abfragen: Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) • Vertiefung an einem weiteren Datenbankbeispiel: Vertiefungen am Beispiel der Relationenalgebra 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modifizieren eine Datenbankmodellierung (M), • bestimmen Primär- und Sekundärschlüssel (M), • analysieren und erläutern eine Datenbankmodellierung (A), • 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). • erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),
<p>3. Gesellschaftliche Auswirkungen der Nutzung von Datenbanksystemen</p>	<ul style="list-style-type: none"> • 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).

Unterrichtsvorhaben Q2-III

Thema: Endliche Automaten und Formale Sprachen

Leitfragen: Wie kann man endliche Automaten genau beschreiben? Wie können endliche Automaten modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, Automaten und Grammatiken?

Zeitbedarf: 25 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Ausgehend von einem konkreten Anwendungsbeispiel entwickeln die Schülerinnen und Schüler das Modell der Grammatik einer formalen Sprache und das Modell des endlichen Automaten. Die Schülerinnen und Schüler überführen Automaten in verschiedene Darstellungsformen und ermitteln die akzeptierte Sprache eines Automaten (z. B. in Form von regulären Ausdrücken). An einem Beispiel wird ein nichtdeterministischer Akzeptor als Alternative gegenüber einem entsprechenden deterministischen Akzeptor eingeführt. Der Zusammenhang zwischen endlichen Automaten und regulären Grammatiken wird durch die Entwicklung allgemeingültiger Verfahren zur Transformation zwischen Automat und Grammatik dargestellt. Die Unzulänglichkeit endlicher Automaten und regulärer Grammatiken wird an Beispielen verdeutlicht.

Lernmittel / Materialien:

- Arbeitsblätter
- Automaten Simulationsprogramm (z. B. Jflap)

Unterrichtssequenzen	zu entwickelnde Kompetenzen
<p>1. Einführung in Automaten/Grammatiken</p> <ul style="list-style-type: none"> • Grammatiken: Grammatik einer natürlichen Sprache Grammatik einer künstlichen Sprache Idee des Parsens • Automaten: erkennender Automat zu Symbolen einer Sprache Modell des endlichen Automaten Darstellungsformen Sprache eines Automaten als regulärer Ausdruck nichtdeterministische Automaten 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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 endliche Automaten (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M).
<p>2. Zusammenhang zwischen endlichen Automaten und regulären Grammatiken</p> <ul style="list-style-type: none"> • reguläre Grammatik: Definition Anwendungen • Zusammenhang zu endlichen Automaten • Grenzen der endlichen Automaten/regulären Grammatiken 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Grammatiken regulärer Sprachen (A), • modifizieren Grammatiken regulärer Sprachen (M), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • 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 im Anwendungszusammenhang auf (A).