



Schulinternes Curriculum I n f o r m a t i k (SII) Luisenschule - Mülheim an der Ruhr

Vorbemerkungen

Dieses schulinterne Curriculum strukturiert den Kompetenzerwerb im Fach Informatik der gymnasialen Oberstufe (SII) an der Luisenschule. Geordnet nach den Jahrgangsstufen (G8) soll er den Akteuren des Informatikunterrichts - den Lernenden wie Lehrenden gleichermaßen - einen Überblick über die zu vermittelnden, inhaltsbezogenen Kompetenzen verschaffen.

Grundlage dieses Curriculums sind die Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe¹ ab 2013, welche sich wiederum auf die verbindlichen Vorgaben der Lehrpläne für die gymnasiale Oberstufe² stützen.

Aufgrund dieser Vorgaben, im Sinne eines zeitgemäßen Informatikunterrichts und der Propädeutik für eine Ausbildung oder ein wissenschaftliches Studium mit informatischen Elementen, ist der objektorientierte Ansatz in der Programmiersprache Java als durchgängiges Paradigma gewählt.

¹ Siehe: <http://www.standardsicherung.nrw.de/abitur-gost/fach.php?fach=15> (abg. am 27.8.2012)

² Richtlinien und Lehrpläne für die Sekundarstufe II - Gymnasium/Gesamtschule, Informatik, Nordrhein-Westfalen, Frechen 1999

Für die die Qualifizierungsphasen vorbereitende Einführungsphase EF liegt der Schwerpunkt in der

- **Einführung in die objektorientierte Modellierung und Programmierung** mit
 - **Einführung in das OO-Paradigma** (Was ist die OO? Wie sieht man mit der „OO-Brille“?),
 - **OO-Modellierung** (Wie kann man Probleme objektorientiert strukturieren, darstellen und analysieren?) und
 - **OO-Programmierung** (Wie kann man Lösungen für Probleme auf Basis von Modellen objektorientiert umsetzen?).

Die gewählten Schwerpunkte für die Qualifizierungsphasen sind gemäß den Vorgaben:

- **I. Objektorientiertes Modellieren und Implementieren von kontextbezogenen Anwendungen** mit
 - **I.1 Konzepte des objektorientierten Modellierens** und
 - **I.2 Algorithmen und Datenstrukturen**
 - [fakultativ und im Folgenden nicht aufgeführt: **I.3 Modellieren und Implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen**]
- **II. Relationale Datenbanken** sowie
- **III. Endliche Automaten und formale Sprachen.**

Die Themenaufstellung ist chronologisch und organsich sinnvoll angelegt. In der Spalte *ZE* sind die voraussichtlich notwendigen Unterrichtswochen angegeben, die Spalte *Thema* formuliert den Überbegriff der in der Spalte *Inhaltsbezogene Kompetenzen* angegebenen Inhalte. Die aufgelisteten Inhalte sind verbindlich, ermöglichen dem Unterrichtenden zugleich ein Spektrum an Variabilität, um individuell und lerngruppengerecht agieren zu können.

Die Spalte *Verknüpfung zu den Vorgaben* stellt die Inhalte in Bezug zu den genannten Vorgaben und legitimiert diese damit. Die Spalte *Materialien, Medien & Werkzeuge* formuliert Ansätze/ Beispiele für Unterrichtsmaterialien und -gegenstände, z.B. geeignete Software-Projekte, Verweise auf sich empfehlende Lehrwerke oder auch didaktische Hinweise, die sich bei der Umsetzung im Unterricht als nützlich erweisen können.

Abschließend sei bemerkt, dass dieses Curriculum zukünftig noch Anpassungen unterworfen sein kann. Einerseits erscheinen die Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe bislang im jährlichen Rhythmus und können die Obligatorik berühren (z.B. gehört im Gegensatz zu den Vorgaben 2013 der inhaltliche Schwerpunkt „Rekursive Algorithmen“ zu den Vorgaben 2015). Andererseits wurde dieses Curriculum gänzlich neu konzipiert und zukünftige, praktische Unterrichtserfahrungen ziehen gegebenenfalls Optimierungen nach sich.

Im September 2012,

Dipl.-Inform. Christian Wolf, StR
Peter Römer, StD

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
4	Paradigma der Objektorientierung (OO)	<p>Die SuS...</p> <ul style="list-style-type: none"> • begreifen das OO-Paradigma als eine abstrakte und menschenähnliche Sicht auf Systeme • lernen die Elemente Objekt, Klasse, Eigenschaft/ Attribut, Attributswert, Dienst/ Methode (als Anfrage und Auftrag) kennen und gebrauchen • unterscheiden die Beziehungen zwischen Objekten/ Klassen mit kennt-, hat- und ist-Beziehung (bzw. Assoziations-, Aggregations-/ Kompositions- und Vererbungsbeziehung) • differenzieren zwischen Spezialisierung und Generalisierung bei der ist-Beziehung • identifizieren und klassifizieren Objekte aus Problembeschreibungen und erstellen einfache Objekt- und Klassen-Karten 	<p>KLP (Objektorientierter Ansatz):</p> <p>Ein Informatikmodell gewinnen: Probleme eingrenzen und spezifizieren, reduzierte Systeme definieren</p> <p><i>„Bei der Entwicklung einer Problemlösung geht man grundsätzlich von dem Gedanken aus, dass die Realität stets durch Objekte beschrieben wird, die Informationen fest halten und miteinander kommunizieren. Dabei steht der Objektbegriff im Mittelpunkt der Betrachtung.“</i></p> <p>Vorgaben ZABI:</p> <p>I.1 Konzepte des objektorientierten Modellierens</p> <ul style="list-style-type: none"> • Klasse, Objekt, Attribut, Methode, Geheimnisprinzip 	<p>Z.B. OO-Ansatz in Grafik- (geometrische Figuren bestehen aus einfacheren geometrischen Figuren mit Attributen) u. Textverarbeitungssystemen (Texte bestehen aus Wörtern, die aus Zeichen bestehen; jedes Zeichen hat bestimmte Attribute und man kann Operationen auf ihm ausführen).*</p> <p>Z.B. Untersuchung der Hierarchie (Vererbung) von Automobilen („Ein PKW ist ein KFZ ist ein Automobil.“)</p> <p>Z.B. einzelne, geometrische Objekte (z.B. Fenster) mit gleichartigen Attributen aus Grafiken (z.B. Kirche) klassifizieren.*</p>

Jahrgangsstufe:

EF **Q1** **Q2**

* Siehe z.B.: Informatik 1 – Objekte, Strukturen, Algorithmen, Klett-Verlag

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
6	OO-Modellierung (OOM)	<p>Die SuS...</p> <ul style="list-style-type: none"> • analysieren Probleme durch die „OO-Brille“, modellieren und stellen sie mittels UML-Elementen dar • entwerfen Objektdiagramme mit Objektkarten (Attribute und Methoden) und den Beziehungen zwischen den Objekten • erfassen die Notwendigkeit und Nützlichkeit von Klassen- gegenüber Objektdiagrammen und aggregieren Objekt- zu Klassendiagrammen • erstellen Klassendiagramme mit Klassen-Karten (Attribute und Methoden) und den Beziehungen (kennt-, hat- und ist-Beziehung) • benutzen Multiplizitäten in Klassendiagrammen und formulieren deren Bedeutung 	<p>KLP:</p> <p>Ein Informatikmodell gewinnen: Probleme eingrenzen und spezifizieren, reduzierte Systeme definieren:</p> <ul style="list-style-type: none"> • Problemstellungen eingrenzen und Probleme strukturieren • Anforderungen an ein Modell aufstellen • ein reduziertes Modell für die Problemstellung definieren <p>Vorgaben ZABI:</p> <p>I.1 Konzepte des objektorientierten Modellierens</p> <ul style="list-style-type: none"> • Klassendiagramme • Beziehungen zwischen Klassen: (Gerichtete) Assoziation mit Multiplizität, Vererbung 	<p>Notationselemente Objekt-, Klassenkarte, Objekt- u. Klassendiagramm angelehnt an die UML.</p> <p>Editoren zum Erstellen von Diagrammen, z.B. yEd Graph Editor oder auch UMLed.</p> <p>Diverse Modellierungsaufgaben, z.B. Szenarien Fluggesellschaft³ oder Computer.</p> <p>Klassische Modellierungen wie z.B. Bank mit Kunde, Bankdirektor, Konto... oder auch Schule mit Schüler(in), Lehrer(in), Personal...</p>

Jahrgangsstufe:

EF Q1 Q2

³ Siehe z.B. auch: S. 31 A. 1, 2, 3 aus Informatik 3 – Algorithmen, Objektorientierte Programmierung, Zustandsmodellierung, Klett-Verlag

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
6	OO-Programmierung (OOP) in Java	<p>Die SuS...</p> <ul style="list-style-type: none"> • erzeugen Objekte aus gegebenen Klassen und rufen deren Dienste auf • implementieren erste eigene Klassen auf Basis einfacher Modelle und realisieren OOP vielmehr als Definition von Klassen denn Objekten • programmieren Methoden ohne und mit Parametern zur Intra-Objekt-Kommunikation und unterscheiden Anfragen von Aufträgen • benutzen Modifizierer zur Sichtbarkeitskontrolle von Attributen und Methoden • erfassen Kontrollstrukturen zur Ablaufsteuerung von Programmen (z.B. Schleifen und bedingte Anweisungen) • benutzen Kontrollstrukturen⁴ zur Programmierung kleinerer Algorithmen und stellen diese dar 	<p>KLP:</p> <p>Lösungskonzepte in einer Programmiersprache realisieren, überprüfen und weiterentwickeln</p> <p><i>„Bei der objektorientierten Programmierung (OOP) werden Programme als kooperierende Ansammlung von Objekten realisiert. [...]“</i></p> <ul style="list-style-type: none"> • Lösungskonzepte implementieren und testen • Lösungen dokumentieren • Lösungen nach vorgegebenen Kriterien bewerten • Problemlösungen optimieren und weiterentwickeln 	<p>USB-Stick mit den notwendigen, portablen Applikationen⁵ zur Entwicklung: BlueJ, Java-Editor, JDK...</p> <p>Didaktische Entwicklungsbibliotheken wie z.B. GeoFaSC⁶ oder SuM.</p> <p>Software-Projekte⁷ z.B.:</p> <ul style="list-style-type: none"> • Mini-Golf • Dart mit Pfeil und Dartscheibe • Grafischer-LKW • ... <p>Darstellung über Struktogramme (z.B. mit dem Werkzeug Structorizer⁵)/ Aktivitätsdiagramme.</p>

Jahrgangsstufe:

EF **Q1** **Q2**

⁴ Einführung in die Aussagenlogik und boolesch'e Algebra zur Thematisierung von verknüpften Bedingungen u.U. erforderlich

⁵ Siehe Informatik on Stick von T. Hempel, <http://www.tinohempel.de/info/info/loStick/index.html> (abg. am 28.8.2012)

⁶ Siehe GeoFaSC von C. Wolf, <http://www.geofasc.de> (abg. am 28.8.2012)

⁷ Siehe auch: B. Schriek: Informatik mit Java, Band I, Nili-Verlag Werl

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
12	OO-Programmierung (OOP) in Java	<p>Die SuS...</p> <ul style="list-style-type: none"> • implementieren eigene Programme auf Basis erweiterter Modelle und stellen die kennt-/ hat-Beziehung eines Objekts als ein Attribut von einem Klassentypen her • programmieren Methoden ohne und mit Parametern (auch Objekte als Parameter) zur Inter-Objekt-Kommunikation • unterscheiden bei Methodenaufrufen „Call-by-value-“ und „Call-by-reference-Parameter“ • verwenden beim Programmieren die ist-Beziehung bei Klassen zur Spezialisierung einer Klasse und Wiederverwendung von Funktionalität • überschreiben geerbte Methoden, um abgeleitete Objekte auch spezialisiert auf Nachrichten reagieren zu lassen (Polymorphie) 	<p>KLP:</p> <p>Lösungskonzepte in einer Programmiersprache realisieren, überprüfen und weiterentwickeln</p> <p><i>„[...] sodass einzelne Objekte oft bereits durch das Senden geeigneter Nachrichten getestet werden können.</i></p> <p>[...]</p> <p><i>Die Wiederverwendbarkeit von Klassen ermöglicht eine leichte Weiterentwicklung.“</i></p> <p>Daten und Algorithmen abstrahieren</p> <p><i>„Dabei entstehen aus bereits vorhandenen Oberklassen durch Erweitern und Überschreiben neue Unterklassen, deren Objekte auf dieselbe Nachricht unterschiedlich reagieren können (Polymorphie).“</i></p>	<p>Software-Projekte⁸ z.B.:</p> <ul style="list-style-type: none"> • Fliegende Kugeln, bei der Kugeln voneinander abprallen; dazu muss jede Kugel alle anderen kennen und es ist eine Kollisionsbehandlung erforderlich • Ampel/ Disco-Licht, bei dem ein Disco-Licht eine Menge von LED-Bars und eine LED-Bar eine Menge von LEDs hat • Blinking-LED, bei der eine solche eine LED ist, die zusätzlich blinken kann • Windradpark • ...

Jahrgangsstufe:

EF **Q1** **Q2**

8 Siehe auch: B. Schriek: Informatik mit Java, Band I, Nili-Verlag Werl

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
7	OO-Programmierung (OOP) in Java	<p>Die SuS...</p> <ul style="list-style-type: none"> • verwenden beim Programmieren die ist-Beziehung zur Generalisierung von Klassen hin zu einer abstrakten Klasse und Bündelung gemeinsamer Funktionalität • erfassen das Entwurfsmuster der Ereignisbehandlung zur Programmierung interaktiver Programme • implementieren einfache interaktive Programme mit Tasten-/ Maussteuerung und differenzieren zwischen Ereignis-auslöser und -handler 	<p>KLP:</p> <p>Daten und Algorithmen abstrahieren</p> <p><i>„[...] Klassen werden aber nicht nur durch Spezialisierung, sondern auch Generalisierung entwickelt.“</i></p> <p>Lösungskonzepte in einer Programmiersprache realisieren, überprüfen und weiter entwickeln</p> <p><i>„Ein typisches programmtechnisches Verfahren der OOP ist die ereignisgesteuerte Programmierung.“</i></p> <p>Vorgaben ZABI:</p> <p>I.1 Konzepte des objektorientierten Modellierens</p> <ul style="list-style-type: none"> • Abstrakte Klassen, Polymorphie 	<p>Software-Projekte⁹ z.B.:</p> <ul style="list-style-type: none"> • Lokomotive, Personenwagen u. Güterwagen in abstrakter Oberklasse Waggon zusammenfassen • Fahrkartenautomat mit Knöpfen zur Bedienung • Einfacher Taschenrechner mit den Grundoperationen • Computerspielvariante von Ping-Pong oder Snake mit Tastatursteuerung <p>Bei Implementierung einfacher GUIs Gebrauch vom MVC-Entwurfsmuster machen.</p>

Jahrgangsstufe:

EF Q1 Q2

⁹ Siehe auch: B. Schriek: Informatik mit Java, Band I, Nili-Verlag Werl

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
6	Datenstrukturen (DS) - Lineare Strukturen	<p>Die SuS...</p> <ul style="list-style-type: none"> • erfassen zunächst 1- und 2-dim. Felder (Arrays) als neue DS neben den bisher bekannten Variablen • erzeugen einfache 1- und 2-dim. Felder mit primitiven Datentypen sowie Objekten • traversieren 1- und 2-dim. Felder und greifen auf Feldelemente zu • benutzen Felder zum Auflösen von 1:n - Beziehungen (kennt-/ hat-Beziehung) • implementieren für 1-dim. Felder die Suchalgorithmen <i>Lineare Suche</i> und <i>Binäre Suche</i> • realisieren für 1-dim. Felder die Sortieralgorithmen <i>Bubble-Sort</i> und <i>Insertion-Sort</i>¹⁰ 	<p>KLP:</p> <p>Daten und Algorithmen abstrahieren</p> <ul style="list-style-type: none"> • Programmierkonzepte allgemeiner und spezieller Art verstehen und benutzen • Formen des Strukturierens einsetzen • problembezogene Objekte und ihre Wechselwirkungen spezifizieren <p>Vorgaben ZABI:</p> <p>I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Lineare Strukturen ([...] Such- und Sortieralgorithmen für Felder [...]) 	<p>Z.B. können über Probleme, wo viele Einzelvariablen notwendig sind (z.B. das Software-Projekt Disco-Licht) Felder motiviert werden.</p> <p>Felder mit Ganzzahlen können zur Primzahlberechnung mit dem Algorithmus von Eratosthenes verwendet werden. Dabei muss ein Feld mehrfach traversiert und auf dasselbe schreibend wie lesend zugegriffen werden.</p> <p>Z.B. eignet sich das Software-Projekt Demokratie¹¹ zur Thematisierung 2-dim. Felder.</p>

Jahrgangsstufe:

EF Q1 Q2

10 Fakultativ: Behandlung von optimiertem Insertion-Sort mit binärer Suche der Einfügeposition

11 Siehe: B. Schriek: Informatik mit Java, Band II, Nili-Verlag Werl

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
9	Datenstrukturen (DS) - Lineare Strukturen	<p>Die SuS...</p> <ul style="list-style-type: none"> vollziehen die Notwendigkeit einer geeigneteren DS als das Feld für dynamische Szenarien nach erfassen die lineare Liste¹² als dynamische DS und begreifen das Konzept der Referenzen (Verweise) zur Verkettung der Listenelemente wenden die Standardoperationen wie z.B. Einfügen, Suchen und Löschen von Elementen auf der Liste an¹³ wenden die Suchalgorithmen <i>Lineare Suche</i> und <i>Binäre Suche</i> auf der Liste an vollziehen die Sortieralgorithmen <i>Bubble-Sort</i> und <i>Insertion-Sort</i> auf einer Liste nach 	<p>KLP:</p> <p>Daten und Algorithmen abstrahieren</p> <ul style="list-style-type: none"> • Programmierkonzepte allgemeiner und spezieller Art verstehen und benutzen • ... <p>Vorgaben ZABI:</p> <p>I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Lineare Strukturen mit den Akzenten <ul style="list-style-type: none"> ◦ Lineare Liste <ul style="list-style-type: none"> ▪ Anwendung der Standardoperationen ◦ Such- und Sortieralgorithmen für Felder und Listen 	<p>Z.B. kann man die Unzulänglichkeit einer Taxiwarteschlange realisiert mit einem Feld aufzeigen (z.B. erfordert das Einfügen das Verschieben von Elementen).</p> <p>Anwendungskontext für eine konkrete Liste: Druckauftragliste (=Liste) vekettet mehrere Druckaufträge (=Listenelemente), die Dokumente (=Inhalt der Listenelemente) enthalten.</p> <p>Referenzen z.B. anschaulich über Telefonbücher oder das einfache Speichermodell des PC als Tabelle.</p> <p>Überführung der konkreten hin zur allgemeinen Liste.</p>

Jahrgangsstufe:

EF Q1 Q2

12 Einfach oder auch doppelt verkettet.

13 Fakultativ: Teilimplementationen der Standardoperationen.

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
7	Datenstrukturen (DS) - Lineare Strukturen	<p>Die SuS...</p> <ul style="list-style-type: none"> • realisieren mittels einer Liste die DS Schlange (Queue) mit ihrem First-In-First-Out-Organisationsprinzip • wenden an und implementieren die Standardoperationen der Schlange, insbesondere Hinten-Anstellen (Enqueue) und Vorne-Wegnehmen (Dequeue) • realisieren mittels einer Liste die DS Stapel (Stack) mit ihrem Last-In-First-Out-Organisationsprinzip • wenden an und implementieren die Standardoperationen des Stapels, insbesondere Oben-Auflegen (Push) und Oben-Wegnehmen (Pop) 	<p>KLP:</p> <p>Daten und Algorithmen abstrahieren</p> <ul style="list-style-type: none"> • Programmierkonzepte allgemeiner und spezieller Art verstehen und benutzen • ... <p>Vorgaben ZABI:</p> <p>I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Lineare Strukturen mit den Akzenten <ul style="list-style-type: none"> ◦ Schlange und Stapel <ul style="list-style-type: none"> ▪ Anwendung der Standardoperationen ▪ Implementation der Standardoperationen 	<p>Die Implementation einer Schlange/ eines Stapels erfolgt geschickt durch Verwendung eines Listen-Objekts (z.B. durch die hat-Beziehung zu einem Listen-Objekt).</p> <p>Man kann die Operationen der DS Schlange und Stapel durch einfache Delegation an das Listen-Objekt realisieren.</p> <p>Bsp. für Software-Projekte mit...</p> <ul style="list-style-type: none"> • Schlangen <ul style="list-style-type: none"> ◦ Taxiwarteschlange ◦ Wartezimmersimulation* • Stapeln <ul style="list-style-type: none"> ◦ Türme von Hanoi (rekursiv)

Jahrgangsstufe:

EF Q1 Q2

* Siehe: B. Schriek: Informatik mit Java, Band II, Nili-Verlag Werl

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
7	Datenstrukturen - Baumstrukturen	<p>Die SuS...</p> <ul style="list-style-type: none"> • erweitern die lineare Liste mit max. einem Nachfolger zur DS des Binärbaums mit max. zwei Nachfolgern • erfassen das Konzept der Rekursion bzw. rekursiver Funktionen und Algorithmen • definieren Binärbäume rekursiv über das Entwurfsmuster Kompositum • wenden die Standardoperationen Einfügen, Suchen und Löschen auf einem Binärbaum an • vollziehen die Traversierungsalgorithmen Pre-, In- und Post-Order auf einem Binärbaum nach 	<p>KLP:</p> <p>Daten und Algorithmen abstrahieren</p> <ul style="list-style-type: none"> • Programmierkonzepte allgemeiner und spezieller Art verstehen und benutzen • ... <p>Vorgaben ZABI:</p> <p>I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Baumstrukturen mit den Akzenten <ul style="list-style-type: none"> ◦ Binärbaum <ul style="list-style-type: none"> ▪ Anwendung der Standardoperationen ▪ Traversierungsalgorithmen 	<p>Binärbaume (z.B. intuitiv Stamm- oder Ahnenbäume) zunächst als einfache Erweiterung einer Liste zum Anknüpfen an den Lernstand der SuS.</p> <p>Das Konzept der Rekursion ist Voraussetzung für den rekursiven Zugang zu Binärbäumen.</p> <p>Fakultativ können rekursive Abläufe über UML-Sequenzdiagramme veranschaulicht werden.*</p> <p>Das Kompositemmuster* bietet Vorteile zur eleganten, rekursiven Implementierung. Konkrete Anwendung ist z.B. eine Datei- und Ordnerverwaltung.</p>

Jahrgangsstufe:

EF Q1 Q2

* Siehe z.B.: Informatik 4 – Rekursive Datenstrukturen, Softwaretechnik, Klett-Verlag

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
6	Datenstrukturen - Baumstrukturen	<p>Die SuS...</p> <ul style="list-style-type: none"> • sortieren Daten bezüglich eines Schlüsselattributs in einen Binärbaum ein und erhalten die DS des binären Suchbaums • wenden die Standardoperationen Einfügen, Suchen und Löschen auf einem binären Suchbaum an • vollziehen die Notwendigkeit des Umstrukturierens (Rotation) beim Löschen eines Knotens aus einem binären Suchbaum nach, um die Suchbaumeigenschaft zu erhalten • wenden die Traversierungsalgorithmen Pre-, In- und Post-Order auf einem binären Suchbaum an 	<p>KLP:</p> <p>Daten und Algorithmen abstrahieren</p> <ul style="list-style-type: none"> • Programmierkonzepte allgemeiner und spezieller Art verstehen und benutzen • ... <p>Vorgaben ZABI:</p> <p>I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Baumstrukturen mit den Akzenten <ul style="list-style-type: none"> ◦ Binärer Suchbaum <ul style="list-style-type: none"> ▪ Anwendung der Standardoperationen ▪ Traversierungsalgorithmen 	<p>Bsp. für Schlüssel:</p> <ul style="list-style-type: none"> • Ganzzahlen • Zeichen • Zeichenketten (z.B. für ein Wörterbuch) <p>Beim Löschen eines Knotens aus einem binären Suchbaum sind drei Fälle zu betrachten¹⁴.</p> <p>Fakultativ: Zusätzliche Operationen wie Min- und Max-Suche können betrachtet werden.</p> <p>Zugang zu Effizienzbetrachtungen mit Komplexitätsanalyse der Standardoperationen möglich.</p>

Jahrgangsstufe:

EF Q1 Q2

¹⁴ Siehe z.B.: Informatik 4 – Rekursive Datenstrukturen, Softwaretechnik, S. 71, Klett-Verlag

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
8	Endliche Automaten und formale Sprachen	<p>Die SuS...</p> <ul style="list-style-type: none"> • definieren deterministische endliche Automaten (DEA) als Verhaltensmodell, bestehend aus Zuständen, Zustandsübergängen und Aktionen • stellen DEA als Zustandsgraph und Tabelle dar • modellieren kontextbezogene Problemstellungen mit DEA • implementieren einfache DEA und überführen dabei DEA in Programm-entsprechungen (z.B. Aktionen in Methoden) • lernen Akzeptoren als spezielle DEA ohne Ausgabe/ ausgelöste Aktionen kennen • untersuchen einfache reguläre Sprachen und entwickeln Akzeptoren für diese • erzeugen einfache reguläre Sprachen durch reguläre Grammatiken und überführen Akzeptoren in reguläre Grammatiken sowie umgekehrt 	<p>KLP:</p> <p>Algorithmen, Sprachkonzepte und Automatenmodelle beurteilen</p> <p><i>„Weil wesentliche Konzepte der Automatentheorie (Zustände, Eingaben als Nachrichten) aus der OOP bekannt sind, lassen sich Automatenmodelle leicht [...] realisieren. Mit Hilfe der Automaten können [...] Sprachkonzepte beurteilt werden.“</i></p> <p>Vorgaben ZABI:</p> <p>III. Endliche Automaten und formale Sprachen</p> <ul style="list-style-type: none"> • Modellieren kontextbezogener Problemstellungen als deterministische endliche Automaten (DEA) • Darstellung von DEA als Graph und als Tabelle • Formale Sprachen: Reguläre Sprachen und ihre Grammatiken 	<p>Zur Einführung in DEA helfen Alltagsbeispiele (z.B. Baustellenlampe, Fahrkartenautomat, GUIs)</p> <p>Modellierungsbeispiel: Hirte-Wolf-Ziege-Kohlkopf-Problem</p> <p>Für Zustandsdiagramme und zur Simulation von DEA kann das Werkzeug JFLAP¹⁵ eingesetzt werden.</p> <p>Bsp. für reguläre Sprachen:</p> <ul style="list-style-type: none"> • E-Mail-Adressen • URL • Ordner-/ Dateipfade <p>Fakultativ können reguläre Ausdrücke zur kompakten Beschreibung regulärer Sprachen thematisiert werden.</p>

Jahrgangsstufe:

15 Siehe: <http://www.jflap.org> (abg. am 10.09.2012)

EF Q1 Q2

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
12	Relationale Datenbanken	<p>Die SuS...</p> <ul style="list-style-type: none"> • unterscheiden die Begriffe Datenbasis, Datenbankmanagementsystem und Datenbanksystem • motivieren die Erfordernis von Datenbanken zur effizienten Verwaltung großer Datenmengen in Informationssystemen • erstellen Datenmodelle mit Klassendiagrammen/ Entity-Relationship-Diagrammen (Entität, Entitätstyp, Attribut, Relation) • überführen Entity-Relationship-Modelle in Datenbankschemata und erstellen so relationale Datenbanken • führen Abfragen auf Datenbanken mittels der SQL durch • wenden die wichtigsten Operationen der Relationenalgebra (s. Verknüpfung zu den Vorgaben →) an • überführen Datenbanken in die 1. bis 3. Normalform 	<p>Vorgaben ZABI:</p> <p>II. Relationale Datenbanken</p> <ul style="list-style-type: none"> • Modellieren kontextbezogener Problemstellungen als Datenbanken mit dem Entity-Relationship-Modell • Datenbankschemata • Normalisierung: Überführung einer Datenbank in die 1. bis 3. Normalform • Relationenalgebra (Selektion, Projektion, Vereinigung, Differenz, kartesisches Produkt, Umbenennung, Join) • SQL-Abfragen über eine und mehrere verknüpfte Tabellen • Datenschutzaspekte 	<p>Software-Werkzeuge¹⁶:</p> <ul style="list-style-type: none"> • Open-Office/ Libre-Office-Base • MS Access • DB-Designer • MySQL • DIA (ER-Diagr.) <p>Lernsequenzen zu relationalen Datenbanken finden sich z.B. in ¹⁷ und ¹⁸.</p> <p>Software-Projekte wie auch weitere Lernsequenzen finden sich auch in ¹⁹.</p>

Jahrgangsstufe:

EF Q1 Q2

¹⁶ Siehe auch: Informatik on Stick von T. Hempel, <http://www.tinohempel.de/info/info/loStick/index.html> (abg. am 28.8.2012)

¹⁷ Grundlagen der Informatik II, Oldenbourg Verlag

¹⁸ Informatik 2 – Tabellenkalkulationssysteme, Datenbanken, Klett-Verlag

¹⁹ B. Schriek: Informatik mit Java, Band III, Nili-Verlag Werl

ZE	Thema	Inhaltsbezogene Kompetenzen	Verknüpfung zu den Vorgaben	Materialien, Medien & Werkzeuge
6	<i>Wiederholungen zur Abiturvorbereitung</i>			

Jahrgangsstufe:

EF Q1 Q2