

Schulinternes Curriculum im Fach Informatik, Sekundarstufe 2

1. VORBEMERKUNG

Bei der Erstellung eines schulinternen Curriculums für das Fach Informatik stellen sich drei Probleme:

1. Der Lehrplan für die Sekundarstufe 1 stammt aus dem Jahr 1993 und ist damit veraltet. Als er verfasst wurde, kannte kaum jemand das Internet; Windows 95 war noch nicht entwickelt; und Vieles, was heute selbstverständliches Wissen jedes Grundschülers ist (z.B.: wie speichere ich eine Datei?) war damals noch vielen Schülerinnen und Schülern unbekannt. Eine Bezugnahme auf diesen Lehrplan ist, soweit es konkrete Unterrichtsinhalte angeht, daher schwierig. Darüber hinaus fehlt die Kompetenz-Orientierung, die alle neueren Lehrpläne haben.
2. Es gibt für das Fach Informatik in NRW keine für den Lehrplan zugelassenen Lehrwerke. Auch ein Lehrwerk kann deshalb nicht auf dieselbe Weise wie in anderen Fächern die Grundlage für das schulinterne Curriculum bilden.
3. Nicht nur die Inhalte, die auch in anderen Fächern Änderungen unterliegen, sondern auch die Methoden der Informatik wandeln sich mit großer Geschwindigkeit. Auch die Lebenswelt der Schülerinnen und Schüler wandelt sich, hinsichtlich ihres Umgangs mit Computern und neuen Informationsformen, in beinahe jährlichem Rhythmus. Je konkreter Inhalte und Methoden deshalb im Curriculum festgelegt werden, desto schneller veraltet dieses Curriculum auch.

Das vorliegende Curriculum verweist teilweise auf die in Bayern zugelassenen Lehrwerke *Informatik 4: Rekursive Datenstrukturen* und *Informatik 5: Formale Sprachen* (Klett Verlag), die zurzeit in der Qualifikationsphase erprobt werden.

2. GESCHLECHTSSPEZIFISCHE UND FÄCHERÜBERGREIFENDE ASPEKTE

Informatik wird in koedukativen Kursen unterrichtet. Eine Unterstützung der je unterschiedlichen Herangehensweise von Jungen und Mädchen findet im Unterricht der Sekundarstufe 2 beispielsweise auf Ebene der individuellen Förderung, der Zusammensetzung von Gruppen bei Gruppenarbeiten und der regelmäßigen Evaluierung der Ergebnisse von Jungen und Mädchen hinsichtlich geschlechtsspezifischer Unterschiede statt. Die Fachschaft Informatik entsendet Mitglieder zu Fortbildungen, die sich mit geschlechtsspezifischer Förderung beschäftigen (z.B. Fortbildung zur "Roberta-Schule" im Jahr 2010).

Fächertübergreifender Unterricht findet im Differenzierungsunterricht zwar nicht explizit statt, da die Kurszusammensetzung und Blockung dies sehr erschweren würde. Innerhalb des Informatik-Unterrichts werden Inhalte und Themen anderer Fächer jedoch häufig als konkreter Gegenstand verwendet, anhand dessen die informatischen Inhalte erarbeitet werden, z.B.: Explizite Thematisierung der Gruppendynamik bei Software-Projekten (Pädagogik); Analyse von Nachrichten in der Unterrichtsreihe über Binärbäume (Sozialwissenschaften); Mathematische Beweisverfahren bei der Behandlung von Rekursion und Kryptographie (Mathematik).

3. INFORMATIK IN DER SEKUNDARSTUFE 2

Solange keine aktualisierten Richtlinien und Lehrpläne vorliegen, wird nach der Umstellung auf das G8 mit den bisherigen Lehrplänen weitergearbeitet, wobei der Lehrplan der Jahrgangsstufe 11 [G9] auf die Einführungsphase [G8] übertragen wird; analoges gilt für die höheren Jahrgangsstufen.

Die Stundenumfänge in der Qualifikationsphase beziehen sich auf den Grundkurs. Sie sind für den Leistungskurs in etwa zu verdoppeln. Inhaltlich wird im Leistungskurs das Programm des Grundkurses behandelt, allerdings in vertiefter, anspruchsvollerer und umfangreicherer Form. So werden bei den Programmierprojekten im Grundkurs-Bereich viele Elemente bereits vorgegeben oder nur in didaktisch reduzierter Form von den Schülerinnen und Schülern verlangt.

EINFÜHRUNGSPHASE

1. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Objekte und Klassen</p> <ul style="list-style-type: none"> • Was sind Objekte, was sind Klassen? • Interaktion mit Objekten durch Methoden • Datentypen und Parameter • Das Innenleben von Objekten 	<p>Methodischer Schwerpunkt: Einzelarbeit am Computer</p>
<p>Klassendefinitionen</p> <ul style="list-style-type: none"> • Aufbau von Klassendefinitionen: Datenfelder, Konstruktoren und Methoden • Sondierende und verändernde Methoden • Zuweisungen und Ausgabe-Anweisungen • Bedingte Verzweigungen • Datenfelder vs. Parameter vs. lokale Variablen 	<p>Methodischer Schwerpunkt: Selbstständige Lektüre wissenschaftlicher Texte und Dokumentationen</p>
<p>Objektinteraktionen</p> <ul style="list-style-type: none"> • Abstraktion und Modularisierung • Primitive Typen und Objektdatentypen • Objekte erzeugen Objekte und rufen sie auf • Nutzung des Debuggers 	<p>Methodischer Schwerpunkt: Implementieren lernen</p>
<p>Objektsammlungen</p> <ul style="list-style-type: none"> • Bibliotheks-Klassen • Was sind Collections (mit generischen Klassen)? • Mit Collections arbeiten • Sammlungen mit fester und mit flexibler Größe 	<p>Methodischer Schwerpunkt: Partnerarbeit</p>

2. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Bibliotheksklassen nutzen</p> <ul style="list-style-type: none"> • Die Dokumentation der Bibliotheksklassen • Verwendung von Maps • Umgang mit Zeichenketten • Sichtbarkeit von Eigenschaften und Methoden • Klassenvariablen 	<p>Methodischer Schwerpunkt: Englische Texte nutzen</p>
<p>Fehler vermeiden</p> <ul style="list-style-type: none"> • Testen als Grundlage professionellen Programmierens • Tests automatisieren • Manuelle Ausführung • Ausgabeanweisungen für die Fehlerbehebung nutzen • Kommentierung und Programmierstil 	<p>Methodischer Schwerpunkt: Reflektieren des eigenen Arbeitens lernen</p>
<p>Programmierprojekt</p> <ul style="list-style-type: none"> • Lebenszyklus von Software, z.B. XP oder Wasserfall-Modell • Projekt: Selbständiges Erstellen einer Software mit dem Lehrer / der Lehrerin als Kunden 	<p>Methodischer Schwerpunkt: Referate vorbereiten und halten; selbstständige Arbeitsorganisation innerhalb einer Großgruppe</p>

QUALIFIKATIONSPHASE, JAHR 1

1. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Klassen- und Objektdiagramme am Beispiel eines Projekts zur Entwicklung von graphischen Benutzerschnittstellen</p> <ul style="list-style-type: none"> • Verwendung der Java API zum Erschließen unbekannter Klassen und Methoden • Syntax und Semantik von Klassen- und Objektdiagrammen • Dokumentation eines Software-Entwurfs mit Hilfe der UML • Umsetzen von Klassendiagrammen in Java-Code • Kennenlernen der wichtigsten Swing-Klassen • Eigenständiges Programmieren eines Abschlussprojekts 	<p><i>Zentralabitur:</i> I.1 Konzepte des objektorientierten Modellierens</p> <ul style="list-style-type: none"> • Klasse, Objekt, Attribut, Methode, Geheimnisprinzip • Klassendiagramme (Entwurfsdiagramme, Implementationsdiagramme) • Beziehungen zwischen Klassen (Gerichtete) Assoziation mit Multiplizität <p>Methodischer Schwerpunkt: Selbstständige Arbeit über längere Phasen mit Hilfe eines Lehrgangs.</p>
<p>Ereignisgesteuerte Programmierung</p> <ul style="list-style-type: none"> • Kennenlernen des Ereignis-Konzepts in der Programmiersprache Java • Erweiterung kleinerer vorgegebener Programme um Interaktivität • Erweiterung des Abschluss-Projekts der vorherigen Unterrichtsreihe 	<p><i>Zentralabitur:</i> I.1 Konzepte des objektorientierten Modellierens</p> <ul style="list-style-type: none"> • Abstrakte Klassen, Vererbung, Polymorphie <p>Methodischer Schwerpunkt: Stationenlernen mit umfangreicher Leistungsdifferenzierung</p>
<p>Die Datenstruktur Stack</p> <ul style="list-style-type: none"> • Entwicklung eines Taschenrechner-Modells • Kennenlernen der Datenstruktur Stack • Umsetzen eines Taschenrechners mit Hilfe von Stacks • Programmier- und Modellier-Übungen zum Stack 	<p><i>Zentralabitur:</i> I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Lineare Strukturen mit den Akzenten Schlange und Stapel (Anwendung der Standardoperationen, Implementation der Standardoperationen), Lineare Liste (Anwendung der Standardoperationen), Such- und Sortieralgorithmen (Lineare Suche, Sortieren durch direktes Einfügen, Quicksort (nur Leistungskurs)) <p>Methodischer Schwerpunkt: Arbeitsteilige Gruppenarbeit</p>
<p>Verlustfreie Datenkompression</p> <ul style="list-style-type: none"> • Theorie der Informationsübermittlung • Huffman- und Lauflängencodierung 	<p>Methodischer Schwerpunkt: Nutzbarmachen von Methoden aus dem Fach Mathematik für die Informatik</p>

<ul style="list-style-type: none"> • Die Datenstruktur "Baum" • Arithmetische Codierung 	
<p>Implementierung eines Lauflängen-Komprimierers mit Hilfe der Datenstruktur Queue</p> <ul style="list-style-type: none"> • Die Datenstruktur Queue (Syntax und Semantik) • Übungen zur Verwendung von Queues • Projekt: Modellierung und Implementierung eines Lauflängen-Codierers 	<p><i>Zentralabitur:</i> I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Lineare Strukturen mit den Akzenten Schlange und Stapel (Anwendung der Standardoperationen, Implementation der Standardoperationen) <p>Methodischer Schwerpunkt: Projektarbeit</p>
<p>Implementierung eines Huffman-Codierers mit Hilfe der Datenstruktur Tree</p> <ul style="list-style-type: none"> • Die Datenstruktur "Tree" nach Vorgabe des Zentralabiturs • Programmier-Übungen zum Tree • Projekt: Implementierung eines Huffman-Codierers 	<p><i>Zentralabitur:</i> I.2 Baumstrukturen mit den Akzenten</p> <ul style="list-style-type: none"> • Binärbaum (Anwendung der Standardoperationen, Traversierungsverfahren) • Geordneter Baum als Spezialfall des Binärbaums (Anwendung der Standardoperationen, Implementation von insertItem und searchItem (nur Leistungskurs)) <p>Methodischer Schwerpunkt: Projektarbeit</p>

2. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Backtracking und Rekursion am Beispiel des Rucksack-Problems</p> <ul style="list-style-type: none"> • Prinzip der Rekursion, Übungen zur Rekursion • Die Klasse der Backtracking-Algorithmen, mit Beispielen und Übungen • Theoretische Lösung des Rucksack-Problems • Implementierung eines Rucksack-Packers • Nicht-Backtracking-Lösungen des Rucksack-Problems 	<p><i>Zentralabitur:</i> I.2 Baumstrukturen mit den Akzenten</p> <ul style="list-style-type: none"> • Binärbaum (Anwendung der Standardoperationen, Traversierungsalgorithmen) <p>Methodischer Schwerpunkt: Analyse und Modellierung von Algorithmen in Gruppenarbeit</p>
<p>Sortieralgorithmen</p> <ul style="list-style-type: none"> • Insertion Sort als Beispiel für einen primitiven Sortieralgorithmus • Implementierung von Insertion Sort auf Basis eines Arrays • Die Datenstruktur Liste in Form der Vorgabe für das Zentralabitur • Implementierung von Insertion Sort auf Basis einer Liste • Quicksort als Beispiel für einen leistungsfähigen Sortieralgorithmus • Nur Leistungskurs: Implementierung von Quicksort auf Basis einer Liste 	<p><i>Zentralabitur:</i> I.2 Datenstrukturen</p> <ul style="list-style-type: none"> • Such- und Sortieralgorithmen (Lineare Suche, Sortieren durch direktes Einfügen, Quicksort (nur Leistungskurs))
<p>Suchalgorithmen</p> <ul style="list-style-type: none"> • Lineare und binäre Suche • Suchen mit Hilfe von binären Bäumen • Die Datenstruktur Binary Tree in Form der Vorgabe des Zentralabiturs • Programmier-Übungen mit der Klasse Binary Tree 	<p><i>Zentralabitur:</i> I.2 Lineare Datenstrukturen</p> <ul style="list-style-type: none"> • Such- und Sortieralgorithmen (Lineare Suche, Sortieren durch direktes Einfügen, Quicksort (nur Leistungskurs)) <p><i>Zentralabitur:</i> I.2 Baumstrukturen mit den Akzenten</p> <ul style="list-style-type: none"> • Geordneter Baum als Spezialfall des Binärbaums (Anwendung der Standardoperationen, Implementation von insertItem und searchItem (nur Leistungskurs))
<p>Graphentheorie</p> <ul style="list-style-type: none"> • Die Datenstruktur Graph • Gewichtete Graphen • Darstellung von Graphen am Rechner (Adja- 	<p><i>Zentralabitur:</i> I.2 Ungerichteter gewichteter Graph (nur Leistungskurs) mit den Akzenten</p> <ul style="list-style-type: none"> • Anwendung der Standardoperationen • Suche des kürzesten Weges zwischen zwei Knoten: Backtracking, Dijkstra-Al-

<p>zenzliste und Adjazenzmatrix)</p> <ul style="list-style-type: none">• Projekt: Ein Navigations-System verstehen und (nur Leistungskurs) implementieren• Nur Leistungskurs: Implementierung fortgeschrittener Graphenalgorithmen (Traversierung, kürzeste Wege, minimaler Spannbaum)	<p>gorithmus</p>
<p>Schwerpunkt nach Interessen der Schülerinnen und Schüler</p>	

QUALIFIKATIONSPHASE, JAHR 2

1. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Deterministische Endliche Automaten</p> <ul style="list-style-type: none"> • Transduktoren und Akzeptoren • Darstellungsformen endlicher Automaten: Tabelle, mathematische Notation, Diagramm • Umsetzung endlicher Automaten mit Java • Automaten in Reguläre Ausdrücke umformen • Reguläre Sprachen zur Erzeugung von Worten, die zu Endlichen Automaten "passen" • Modellierungs- und Programmierübungen • Nur Leistungskurs: Entwicklung eines Parsers für Mini-Java 	<p><i>Zentralabitur:</i> III. Endliche Automaten und formale Sprachen</p> <ul style="list-style-type: none"> • Modellieren kontextbezogener Problemstellungen als endliche Automaten • Darstellung von deterministischen endlichen Automaten als Graph und als Tabelle • Akzeptor als spezielle Form des endlichen Automaten • Formale Sprachen: Reguläre Sprachen • Entwicklung eines Parsers für eine einfache formale Sprache (nur Leistungskurs)
<p>Kryptographie</p> <ul style="list-style-type: none"> • Klassische Verschlüsselungsverfahren: Monoalphabetische Substitution und Vigenère • Häufigkeitsanalyse • Enigma als Ende klassischer Kryptographie • Entschlüsseln echter Enigma-Funksprüche • RSA als Beispiel für ein modernes Verschlüsselungs-Verfahren • Das Schlüsselübermittlungsproblem und seine Lösung (Diffie-Hellmann) 	<p><i>Zentralabitur:</i> I.3 Modellieren und Implementieren kontextbezogener Problemstellungen als Netzerkanwendungen</p> <ul style="list-style-type: none"> • Kryptografie (Symmetrische Verschlüsselungsverfahren (Cäsar, Vigenère); Asymmetrische Verschlüsselungsverfahren (RSA); Schlüsselaustausch (Diffie-Hellmann))
<p>Datenbanken</p> <ul style="list-style-type: none"> • Definitionen zu Datenbanken • Modellierung von Datenstrukturen • Normalformen • SQL-Abfragen • Verwendung von MySQLAdmin • Projekt: Informationen Datenbank gewinnen • Nur Leistungskurs: Datenbanken mit Java abfragen 	<p><i>Zentralabitur:</i> II. Relationale Datenbanken</p> <ul style="list-style-type: none"> • Modellieren kontextbez. Problemstellungen als Datenbanken mit ER-Modell • Normalisierung • Realisierung in einem Datenbanksystem • Relationenalgebra (Select, Project, Join) • SQL-Abfragen über eine und mehrere verknüpfte Tabellen • Datenschutzaspekte

2. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Client-Server-Programmierung</p> <ul style="list-style-type: none"> • Der TCP-IP-Stack • Netzwerk-Services • Verteilte Algorithmen, z.B.: Routing im Internet; Primzahl-Ermittlung, Cloud Computing • Implementieren eines einfachen Client-Server-Dienstes 	<p><i>Zentralabitur:</i> I.3 Modellieren und Implementieren kontextbezogener Problemstellungen als Netzerkanwendungen</p> <ul style="list-style-type: none"> • Netzwerkprotokolle • Client-Anwendungen • Client-Server-Anwendungen
<p>Wiederholung und Übungen</p>	