

Schulinternes Curriculum im Fach Informatik, Sekundarstufe 2

1. 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.

Fächerübergreifender 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).

2. INFORMATIK IN DER SEKUNDARSTUFE 2

Die Stundenumfänge in der Qualifikationsphase beziehen sich auf den Grundkurs. Sie sind für den Leistungskurs in etwa zu verdoppeln. Inhaltlich wird - bis auf ausgewählte Themenbereiche wie z. B. Graphentheorie - 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 	<ul style="list-style-type: none"> • Methodischer Schwerpunkt: Einzelarbeit am Computer • Einführungsprojekte mit BlueJ
<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 	<ul style="list-style-type: none"> • Methodischer Schwerpunkt: Selbstständige Lektüre wissenschaftlicher Texte und Dokumentationen • Projekte mit Greenfoot und BlueJ
<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 	<ul style="list-style-type: none"> • Methodischer Schwerpunkt: Implementieren lernen • Java-Projekte mit BlueJ
<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 	<ul style="list-style-type: none"> • Methodischer Schwerpunkt: Partnerarbeit • Java-Projekte mit BlueJ

2. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Bibliotheksclassen nutzen</p> <ul style="list-style-type: none"> • Die Dokumentation der Bibliotheksclassen • 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 • 3D-Modellierung und Programmierung: GLOOP-Projekte • Projekt: Selbständiges Erstellen einer Software mit dem Lehrer / der Lehrerin als Kunden 	<p>Methodischer Schwerpunkt:</p> <ul style="list-style-type: none"> • Referate vorbereiten und halten; selbstständige Arbeitsorganisation innerhalb einer Großgruppe • Java-Projekte mit GLOOP

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 BinaryTree</p> <ul style="list-style-type: none"> • Die Datenstrukturen "Tree", "BinaryTree" und "BinarySearchTree" nach Vorgabe des Zentralabiturs • Programmier-Übungen zu BinaryTree und BinarySearchTree • Projekt: Implementierung eines Huffman-Codierers 	<p><i>Zentralabitur:</i> I.3 Baumstrukturen mit den Akzenten</p> <ul style="list-style-type: none"> • Binärbaum (Anwendung der Standardoperationen, Traversierungsalgorithmen) • Geordneter Baum (Binärer Suchbaum) als Spezialfall des Binärbaums (Anwendung der Standardoperationen, Implementation (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 • 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.3 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"> • InsertionSort und SelectioSort als Beispiele für primitive Sortieralgorithmen • Implementierung von InsertionSort und SelectionSort auf Basis eines Arrays • Die Datenstruktur Liste in Form der Vorgabe für das Zentralabitur • Implementierung von InsertionSort und SelectionSort auf Basis einer Liste • Quicksort als Beispiel für einen leistungsfähigen Sortieralgorithmus • Nur Leistungskurs: Implementierung von Quicksort auf Basis einer Liste 	<p><i>Zentralabitur:</i> II.1 Algorithmen</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementation von Algorithmen • Such- und Sortieralgorithmen (Lineare Suche, Sortieren durch direktes Einfügen, Quicksort (nur Leistungskurs)) • Modellierung: Struktogramme
<p>Suchalgorithmen</p> <ul style="list-style-type: none"> • Lineare und binäre Suche • Suchen mit Hilfe von binären (Such)bäumen • Die Datenstruktur BinarySearchTree in Form der Vorgabe des Zentralabiturs • Programmier-Übungen mit der Klasse BinarySearchTree 	<p><i>Zentralabitur:</i> I.2 Lineare Datenstrukturen</p> <p><i>Zentralabitur:</i> II.2 Algorithmen in ausgewählten informatischen Kontexten</p> <ul style="list-style-type: none"> • Such- und Sortieralgorithmen (Lineare Suche, Sortieren durch direktes Einfügen, Quicksort (nur Leistungskurs)) <p><i>Zentralabitur:</i> I.3 Nicht-lineare Strukturen</p> <ul style="list-style-type: none"> • Geordneter Baum als Spezialfall des Binärbaums
<p>Graphentheorie</p> <ul style="list-style-type: none"> • Die Datenstruktur Graph • Gewichtete Graphen • Darstellung von Graphen am Rechner (Adjazenzliste und Adjazenzmatrix) 	<p><i>Zentralabitur:</i> I.3 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-Algorithmus

<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)	
Schwerpunkt nach Interessen der Schülerinnen und Schüler	

QUALIFIKATIONSPHASE, JAHR 2

1. Halbjahr

Inhaltliche Schwerpunkte	Bezüge zur Obligatorik, Methodische Schwerpunkte, Kompetenzen
<p>Endliche Automaten und Kellerautomaten</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" • Deterministische und nichtdeterministische endliche Automaten • Nur Leistungskurs: Nichtdeterministische Kellerautomaten • Nur Leistungskurs: Grammatiken regulärer vs kontextfreier Sprachen • Produktionen mit Epsilon • Modellierungs- und Programmierübungen • Nur Leistungskurs: Scanner, Parser und Interpreter für eine reguläre Sprache • Möglichkeiten und Grenzen von Automaten und formalen Sprachen 	<p><i>Zentralabitur:</i> III. Formale Sprachen und Automaten</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 • Kellerautomaten und kontextfreie Sprachen (nur Leistungskurs) • 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> II.2 Modellieren und Implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen</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 	<p><i>Zentralabitur:</i> I.4 Datenbanken</p> <p><i>Zentralabitur:</i> III.1, SQL</p> <ul style="list-style-type: none"> • Modellieren kontextbez. Problemstellungen

<ul style="list-style-type: none">• Normalformen• SQL-Abfragen• Verwendung von MySQLAdmin• Projekt: Informationen Datenbank gewinnen• Datenbanken mit Java abfragen; Klassen DataBaseConnector, QueryResult• Sicherheit	<p>gen als Datenbanken mit ER-Modell</p> <ul style="list-style-type: none">• 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, Netzwerkprotokolle • Netzwerk-Services • Verteilte Algorithmen, z.B.: Routing im Internet; Primzahl-Ermittlung, Cloud Computing • Nur Leistungskurs: Implementieren eines einfachen Client-Server-Dienstes mit den Klassen Connection, Client, Server • Modellieren und Implementieren kontextbezogener Problemstellungen als Netzwerkanwendungen (Netzwerkprotokolle, Client-Anwendungen, Client-Server-Anwendungen) • Einzelrechner und Rechnernetzwerke • Sicherheit 	<p><i>Zentralabitur:</i> II.2 Algorithmen in ausgewählten informatischen Kontexten</p> <ul style="list-style-type: none"> • Nur Leistungskurs: Algorithmen zur Kommunikation in Netzwerken (Klassen Connection, Client, Server) <p><i>Zentralabitur:</i> IV.1 Informatiksysteme</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Nutzung von Informatiksystemen • Sicherheit
<p>Informatik, Mensch und Gesellschaft</p> <ul style="list-style-type: none"> • Grundprinzipien des Datenschutzes (Verbot mit Erlaubnisvorbehalt, Erforderlichkeit) • Grenzen der Automatisierung • Technische "Zukunftsszenarien" 	<p>Methodischer Schwerpunkt: Referate vorbereiten und halten; selbstständige Arbeitsorganisation</p>
<p>Wiederholung und Übungen</p>	