

Schulinterner Lehrplan für das Fach Informatik der gymnasialen Oberstufe am Friedrich-Bährens-Gymnasium

Vorbemerkung: Am Friedrich-Bährens-Gymnasium wird derzeit das Fach Informatik in der Qualifikationsphase ausschließlich als Grundkurs erteilt.

1. Unterrichtsvorhaben in der Einführungsphase

- E-I: Allgemeine Einführung
- E-II: Grundkonzepte der Objektorientierung
- E-III: Kontrollstrukturen
- E-IV: Objektbeziehungen
- E-V: Datenstruktur Feld
- E-VI: Sortier- und Suchverfahren (ohne Implementierung)
- E-VII: Vererbung (ohne Polymorphie und abstrakte Klassen)
- E-VIII: Digitalisierung, Datenschutz und Geschichte der digitalen Datenverarbeitung

2. Zuordnung der konkretisierten Kompetenzerwartungen bis zum Ende der Einführungsphase zu den Unterrichtsvorhaben

3. Unterrichtsvorhaben in der Qualifikationsphase

- Q-I: Vererbung (mit Polymorphie und abstrakten Superklassen)
- Q-II: Rekursion
- Q-III: Such- und Sortierverfahren (mit Implementierung)
- Q-IV: Lineare Strukturen (Schlange, Liste, Stapel)
- Q-V: Datenbanken
- Q-VI: Sicherheit und Datenschutz in Netzstrukturen
- Q-VII: Nichtlineare Strukturen (binäre Bäume)
- Q-VIII: Formale Sprachen (Automaten und Grammatiken)
- Q-IX: Funktionsweise eines Computers und Grenzen der Automatisierbarkeit

4. Zuordnung der konkretisierten Kompetenzerwartungen bis zum Ende der Qualifikationsphase zu den Unterrichtsvorhaben

5. Grundsätze der Leistungsbewertung

1. Unterrichtsvorhaben in der Einführungsphase

Hinweis: Kompetenzbereiche sowie inhaltliche Schwerpunkte und zugehörige Inhaltsfelder, die in nahezu allen Unterrichtsvorhaben der Einführungsphase relevant sind, werden hier aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt.

E-I: Allgemeine Einführung

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner
- Dateisystem
- Internet
- Einsatz von Informatiksystemen

Inhalte	Beispiele, Ergänzungen, Hinweise
1. Darstellen von Informationen und Deuten von Daten Informatik als Wissenschaft der systematischen Verarbeitung von Informationen; Informationen und ihre Darstellung; Daten und ihre Deutung; Binärdarstellung von Daten	<i>Beispiel:</i> Hochhausbeleuchtung zur Darstellung von Fußballergebnissen <i>Beispiel:</i> Analysieren eines maschinennahen Programms, das den Wert eines Terms der Form $a \times b + a \times c$ berechnet.
2. Aufbau und Arbeitsweise eines Einzelrechners (von Neumann) Grundkomponenten und schematische Darstellung eines von-Neumann-Rechners; Phasen während der Ausführung eines maschinennahen Programms durch einen von-Neumann-Rechner; Analysieren eines einfachen maschinennahen Programms	
3. Verwenden der Schulrechner und des Schulnetzwerks Erläuterungen zum Schulnetzwerk und Vereinbaren von Regeln zur Benutzung der Schulrechner (z.B. Anmeldung am Server, Speichern von Dateien, Austausch von Dateien, usw.); Vereinbaren von Regeln zum Datenaustausch und zur Kommunikation über das Internet	

E-II: Grundkonzepte der Objektorientierung**Zentrale Kompetenzen:**

- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Objekte und Klassen Objekte und ihre Attribute; Objektdiagramme; Methoden von Objekten; Klassen und ihre schematische Darstellung (Klassendiagramm); Einführung in BlueJ (Objekte erzeugen; Methoden ohne und mit Parametern von Objekten aufrufen; Inspizieren von Objekten - Objektinspektor; Direkteingabe und Punktnotation; Analysieren von Fehlermeldungen) Datentypen und ihre Bedeutung; Rückgabewerte von Methoden; Signatur einer Methode</p>	<p><i>Beispiel:</i> Analysieren und Erstellen von Graphiken aus geometrischen Figuren unter Verwendung eines einfachen objektorientierten Graphikprogramms</p> <p><i>Beispiel:</i> Buchungssystem für Eintrittskarten eines Opernhauses</p> <p><i>Beispiel:</i> Verwalten eines Sparkontos</p>
<p>2. Implementieren von Klassen Äußerer Rahmen und Innenteil des Quelltextes einer Klasse; Deklarieren von Attributen; privates Zugriffsrecht für Attribute; Variablenkonzept; Übersetzen eines Quelltextes (Compiler und Interpreter; Bytecode und Maschinencode; Plattformunabhängigkeit von Java); Kommentare in Quelltexten; Implementieren von Konstruktoren (ohne und mit Parametern); Wertzuweisung; Default-Konstruktor; Implementieren von Methoden (ohne und mit Parametern): Anfragen und Getter-Methoden; <code>return</code>-Anweisung; Aufträge und Setter-Methoden; Rückgabotyp <code>void</code>; lokale Variablen in Blöcken; arithmetische Operatoren in Java; Ausgabeanweisung Klassendokumentation in Schriftform und in Quelltexten (Javadoc)</p>	

E-III: Kontrollstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Nutzen einer Klassenbibliothek – Referenzattribute – Sequenz Untersuchen der Dokumentation der Klassen von <code>javakarol</code>; Überladen von Methoden; Analysieren eines Beispielprogramms (Import von Klassen aus der Klassenbibliothek; Deklarieren und Erzeugen von Objekten; Methodenaufruf durch Punktnotation); Referenzattribute; Algorithmus; Struktogramm; Sequenz</p>	<p><i>Beispiel:</i> Verwenden der Klassenbibliothek <code>javakarol</code> und Nutzen der zugehörigen Dokumentation. Beispiele für zu entwickelnde Programme:</p> <ul style="list-style-type: none"> • Karol soll am Rand seiner Welt eine Mauer bauen. Die Welt hat zunächst eine festgelegte Größe und kann später z.B. durch Parameter eine flexible Größe haben • Karol soll um ein Grundstück patrouillieren
<p>2. Schleifen Zählschleife; private Hilfsmethoden und interner Methodenaufruf; Schleife mit Eingangsbedingung; Negieren von Wahrheitswerten; Schleife mit Ausgangsbedingung; Endlosschleife; Umwandeln der Schleifenarten; Verschachteln von Schleifen; Speichern von Rückgabewerten in lokalen Variablen; Vergleichsoperatoren in Java</p>	
<p>3. Verzweigungen – boolesche Ausdrücken einseitige und zweiseitige bedingte Anweisung; Verschachteln bedingter Anweisungen; logische Operatoren zum Verknüpfen von Wahrheitswerten; boolesche Ausdrücke; de Morgan'sche Regeln; Mehrfachauswahl</p>	

E-IV: Objektbeziehungen**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Algorithmen zum Suchen und Sortieren
- Syntax und Semantik einer Programmiersprache

Inhalte	Beispiele, Ergänzungen, Hinweise
1. Gerichtete Assoziation – Objektbeziehungen herstellen gerichtete Assoziation zwischen Objekten; Multiplizitäten; Entwurfs- und Implementationsdiagramm; Realisieren einer gerichteten Assoziation durch Referenzattribute; Zuweisen eines Referenzziels (Setter-Methode, Konstruktor); null-Referenz	<i>Beispiel:</i> Entwerfen und Implementieren eines Chat-Systems. Zunächst wird dabei ein starres System mit zwei sich fest zugeordneten Chatpartnern realisiert, anschließend erfolgt eine Erweiterung auf ein System mit einer zentralen Chatverwaltung, die beliebig viele Chatter verwalten kann.
2. Kommunikation zwischen Objekten – Beziehungen verwenden Anbieter und Nutzer von Diensten; Nutzen eines Dienstes durch Aufruf einer öffentlichen Methoden eines referenzierten Objekts	
3. Datenkapselung und Zugriffsrechte privates Zugriffsrecht für Attribute; kontrollierte Zugriffsmöglichkeit auf Attribute durch öffentliche Getter- bzw. Settermethoden; Prinzip der Datenkapselung; Zugriffsmöglichkeit auf private Ressourcen durch Objekte derselben Klasse	
4. Datenansammlung – die Klasse ArrayList Realisieren einer gerichtete Assoziation mit der Multiplizität 0..* durch eine Datenansammlung; Java-Klassenbibliothek (Java-API); Nutzen der Klasse ArrayList; Selbstreferenz this; sequentielle Suche; Dokumentation der Methoden der Klassen des Projekts	

E-V: Datenstruktur Feld

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Eindimensionale Felder Datenansammlung mit fester Größe (Array): Grundprinzip; Array-Variablen deklarieren, Array-Objekte erzeugen; Array-Objekte benutzen; Erzeugen von Zufallszahlen: die Klasse Random und ihre Methoden; das Attribut length eines Array-Objekts; Verwenden von Feldern in Implementationsdiagrammen; Dateneingabe über die Konsole: die Klasse Scanner und ihre Methoden; Minimumsuche in einem Feld</p>	<p><i>Beispiel:</i> Simulation zum Rosinenproblem</p> <p><i>Beispiel:</i> Simulation der Ziehung der Lottozahlen</p> <p><i>Beispiel:</i> Entwickeln und Implementieren des Ratespiels „Zahlenraten“</p> <p><i>Beispiel:</i> Untersuchen potentiell magischer Quadrate</p>
<p>2. Zweidimensionale Felder Grundprinzip, Deklaration, Erzeugung und Elementzugriff für zweidimensionale Felder; Anwenden an einem Beispiel</p>	

E-VI: Sortier- und Suchverfahren

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Algorithmen

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Algorithmen zum Suchen und Sortieren

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Sortieren Analysieren und Anwenden eines vorgegebenen Sortieralgorithmus (InsertionSort); Erarbeiten weiterer Sortierstrategien anhand geeigneter Materialien; Formulieren und Präsentieren der erarbeiteten Sortierstrategien; Entwerfen von Algorithmen zu den erarbeiteten Sortierstrategien (BubbleSort, SelectionSort); Eigenschaften von Sortierverfahren (in-place-Verfahren, stabile Verfahren); Erarbeiten von Kriterien zum Vergleich der Effizienz der Sortierverfahren; Vergleichen der Effizienz der behandelten Sortierverfahren (anhand eines Beispiels und anhand empirischer Messungen)</p>	<p><i>Beispiel:</i> Sortieren von mit Zahlen beschrifteten Karten unter Beachtung bestimmter Vorgaben (z.B. es dürfen nur jeweils zwei Karten aufgedeckt und ihre Zahlenwerte miteinander verglichen werden)</p>
<p>2. Suchen Analysieren und Anwenden eines vorgegebenen Suchverfahrens (binäre Suche); Erarbeiten der wesentlichen Voraussetzung, unter der die binäre Suche durchgeführt werden kann</p>	

E-VII: Vererbung**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Syntax und Semantik einer Programmiersprache

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Projektentwicklung ohne Verwendung der Vererbung Entwerfen einer objektorientierten Modellierung ohne Vererbung, wobei Teile des Quelltextes mehrerer Klassen identisch sind; Analysieren und Beurteilen des Quelltextes der Klassen unter der Voraussetzung, dass weitere Klassen „gleichen Typs“ zum Modell hinzugefügt werden sollen</p>	<p><i>Beispiel:</i> Entwerfen und Implementieren von Aspekten eines WebShops</p> <p><i>Beispiel:</i> Entwerfen und Implementieren von Aspekten einer Kontoverwaltung für Konten unterschiedlicher Kontotypen</p>
<p>2. Projektentwicklung unter Verwendung der Vererbung Vererbungsbeziehung zwischen Klassen; Schematische Darstellung der Vererbung in Entwurfs- und Implementationsdiagrammen; Generalisierung und Spezialisierung; Vererbungshierarchie; Umsetzung des Vererbungskonzepts in Java; Zugriffsmodifikator <code>protected</code>; Aufruf des Konstruktors der Superklasse mittels <code>super (. . .)</code>; Substitutionsprinzip; die Klasse <code>Object</code> als Superklasse aller Klassen; falls es sich in früheren Unterrichtsvorhaben noch nicht ergeben hat: Kennenlernen und Verwenden des Debuggers von BlueJ</p>	

E-VIII: Digitalisierung, Datenschutz und Geschichte der digitalen Datenverarbeitung

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Digitalisierung
- Internet
- Wirkungen der Automatisierung
- Geschichte der automatischen Datenverarbeitung

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Entwickeln von Präsentationen zu ausgewählten informatischen Themenbereichen</p> <p>Selbstständiges Recherchieren z.B. zu folgenden Themen:</p> <ul style="list-style-type: none"> • Stellenwertsysteme und Rechnen in Stellenwertsystemen • Kodieren von Zahlen, Texten und Bildern • Geschichte der Digitalisierung • Geschichte der Kryptographie • Auswirkungen der Digitalisierung: Datenschutz <p>Eingrenzen der Themen; Erstellen von Plakatpräsentationen; Erarbeiten grundlegender Begriffe des Datenschutzes; Bewerten von Fallbeispielen aus dem Themenbereich „Datenschutz“ basierend auf dem Bundesdatenschutzgesetz</p>	<p><i>Beispiel:</i> Ausstellung zu informatischen Themen</p> <p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen</p>

2. Zuordnung der konkretisierten Kompetenzerwartungen bis zum Ende der Einführungsphase zu den Unterrichtsvorhaben

Kompetenzerwartungen	Unterrichtsvorhaben							
	E-I	E-II	E-III	E-IV	E-V	E-VI	E-VII	E-VIII
Inhaltsfeld 1: Daten und ihre Strukturierung								
<u>Inhaltlicher Schwerpunkt:</u> - Objekte und Klassen								
<i>Objekte und Klassen</i> Die Schülerinnen und Schüler								
- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),		X		X	X		X	
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),		X		X	X		X	
- modellieren Klassen unter Verwendung von Vererbung (M),							X	
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),		X		X	X		X	
- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),		X		X	X		X	
- stellen den Zustand eines Objekts dar (D),		X					X	
- stellen die Kommunikation zwischen Objekten grafisch dar (M),				X				
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),		X		X	X		X	
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D),		X		X			X	
- analysieren und erläutern eine objektorientierte Modellierung (A),			X	X	X		X	
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).		X	X	X	X		X	

Kompetenzerwartungen	Unterrichtsvorhaben															
	E-I	E-II	E-III	E-IV	E-V	E-VI	E-VII	E-VIII								
Inhaltsfeld 2: Algorithmen																
<u>Inhaltliche Schwerpunkte:</u>																
- Analyse, Entwurf und Implementierung einfacher Algorithmen																
- Algorithmen zum Suchen und Sortieren																
<i>Analyse, Entwurf und Implementierung einfacher Algorithmen</i>																
Die Schülerinnen und Schüler																
- analysieren und erläutern einfache Algorithmen und Programme (A),											X			X	X	
- modifizieren einfache Algorithmen und Programme (I),											X				X	
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),											X		X	X		
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),											X		X			
- testen Programme schrittweise anhand von Beispielen (I).			X	X	X		X									
<i>Algorithmen zum Suchen und Sortieren</i>																
Die Schülerinnen und Schüler																
- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D),												X		X		
- entwerfen einen weiteren Algorithmus zum Sortieren (M),														X		
- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A).														X		

Kompetenzerwartungen	Unterrichtsvorhaben							
	E-I	E-II	E-III	E-IV	E-V	E-VI	E-VII	E-VIII
Inhaltsfeld 3: Formale Sprachen und Automaten								
<u>Inhaltlicher Schwerpunkt:</u> - Syntax und Semantik einer Programmiersprache								
<i>Syntax und Semantik einer Programmiersprache</i> Die Schülerinnen und Schüler								
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),		X	X	X	X		X	
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).		X	X	X	X		X	
Inhaltsfeld 4: Informatiksysteme								
<u>Inhaltliche Schwerpunkte:</u> - Digitalisierung - Einzelrechner - Dateisystem - Internet								
<i>Digitalisierung</i> Die Schülerinnen und Schüler								
- stellen ganze Zahlen und Zeichen in Binärcodes dar (D),								X
- interpretieren Binärcodes als Zahlen und Zeichen (D).								X
<i>Einzelrechner</i> Die Schülerinnen und Schüler								
- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).	X							

Kompetenzerwartungen	Unterrichtsvorhaben							
	E-I	E-II	E-III	E-IV	E-V	E-VI	E-VII	E-VIII
<i>Dateisystem</i> Die Schülerinnen und Schüler								
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).	X	X	X	X	X		X	
<i>Internet</i> Die Schülerinnen und Schüler								
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	X	X	X	X	X		X	X
Inhaltsfeld 5: Informatik, Mensch und Gesellschaft								
<u>Inhaltliche Schwerpunkte:</u> - Einsatz von Informatiksystemen - Wirkungen der Automatisierung - Geschichte der automatischen Datenverarbeitung								
<i>Einsatz von Informatiksystemen</i> Die Schülerinnen und Schüler								
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).	X	X	X	X	X		X	
<i>Wirkungen der Automatisierung</i> Die Schülerinnen und Schüler								
- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).								X
<i>Geschichte der automatischen Datenverarbeitung</i> Die Schülerinnen und Schüler								
- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).								X

3. Unterrichtsvorhaben in der Qualifikationsphase

Hinweis: Kompetenzbereiche sowie inhaltliche Schwerpunkte und zugehörige Inhaltsfelder, die in nahezu allen Unterrichtsvorhaben der Qualifikationsphase relevant sind, werden hier aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt.

Q-I: Vererbung (mit Polymorphie und abstrakten Superklassen)

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Syntax und Semantik einer Programmiersprache

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Konzept der Polymorphie von Methoden Analysieren eines Problems, das durch Verwenden des Konzepts der Polymorphie gelöst werden kann; Vererben und Redefinieren von Methoden; Mechanismus der späten Bindung; Polymorphie von Methoden; Aufrufen von überschriebenen Methoden der Superklasse mittels der Referenz <code>super</code></p>	<p><i>Beispiel:</i> Entwerfen und Implementieren von Aspekten eines WebShops</p> <p><i>Beispiel:</i> Entwerfen und Implementieren von Aspekten einer Kontoverwaltung für Konten unterschiedlicher Kontotypen</p>
<p>2. Abstrakte Klassen und Interfaces Erarbeiten der Grundidee und der Eigenschaften abstrakter Klassen und abstrakter Methoden; Implementieren und Vererben abstrakter Klassen mit abstrakten Methoden; Eigenschaften eines Interface; Vererbung mit Interfaces; Interfaces als Objekttypen</p>	
<p>3. Modellierung mit Vererbung und Assoziation im Vergleich Analysieren einer Situation, bei der die Modellierung mittels Vererbung möglich, aber problematisch ist; Erarbeiten einer alternativen Modellierung mittels Assoziation; Erarbeiten von Kriterien, die eine Modellierung mittels Vererbung nahelegen</p>	

Q-II: Rekursion**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Inhalte	Beispiele, Ergänzungen, Hinweise
1. Analysieren rekursiver Algorithmen Analysieren gegebener Java-Methoden unter den Aspekten: endständige Rekursion, echte Rekursion, Einsatz von Parametern; Untersuchen (mittels BlueJ-Debugger), Darstellen und Erläutern der Abläufe bei der Ausführung rekursiver Methoden	<i>Beispiel:</i> Fakultät <i>Beispiel:</i> Fibonacci-Folge <i>Beispiel:</i> Zeichenketten „umdrehen“; Palindromerkennung
2. Erstellen rekursiver Algorithmen Erstellen rekursiver (auch verzweigte Rekursion) und entsprechender iterativer Algorithmen; Strategie des Problemlösens durch Delegieren; Darstellen der Aufrufhierarchie bei einfacher und verzweigter Rekursion; Effizienzbetrachtung ausgewählter rekursiver Verfahren; Vergleich Rekursion und Iteration; Bearbeiten von Zeichenketten (Methoden der Klasse String)	<i>Beispiel:</i> Die Türme von Hanoi

Q-III: Such- und Sortierverfahren (mit Implementierung)**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Inhalte	Beispiele, Ergänzungen, Hinweise
1. Suchverfahren Untersuchen des rekursiven binären Suchalgorithmus; Vergleich der Effizienz des binären Suchverfahrens und des sequentiellen Suchverfahrens; das Prinzip „Teile und Herrsche“; Implementieren der binären Suche	<i>Beispiel:</i> Felder mit ganzzahligen Werten (aufsteigend geordnet, absteigend geordnet, zufällige Belegung der Komponenten des Feldes). Für die Sortierverfahren kann ein Prototyp zur Verfügung gestellt werden.
2. Sortierverfahren Wiederholen der einfachen Sortierverfahren; Effizienzvergleich der einfachen Sortierverfahren; Erarbeiten eines detaillierten Algorithmus für mindestens ein einfaches Sortierverfahren; Implementieren von mindestens einem einfachen Sortierverfahren; theoretische Aufwandsuntersuchung für mindestens ein einfaches Sortierverfahren (best case, worst case, average case); Entwickeln eines Algorithmus zum QuickSort-Verfahren; Implementieren des QuickSort-Verfahrens; theoretische Aufwandsanalyse des QuickSort-Verfahrens (best case, worst case, average case)	

Q-IV: Lineare Strukturen (Schlange, Liste, Stapel)

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Warteschlange (Queue) Verketteten von Objekten; First-In-First-Out-Prinzip; graphisches Darstellen der Struktur einer Warteschlange; Implementieren der Standardmethoden einer Warteschlange; Trennen von Knoten- und Datenobjekten; Generische Warteschlangen; Erstellen einer Klassenbibliothek mit der generischen Klasse <code>Queue</code>; Dokumentieren der Klasse <code>Queue</code>; Modellieren mit Datenstrukturen (Entwurfs- und Implementationsdiagramm); Verwenden der erstellten Klassenbibliothek im Rahmen eines Softwareprojekts (objektorientierte Analyse, objektorientiertes Design, objektorientierte Programmierung)</p>	<p><i>Beispiel:</i> Patientenwarteschlange in eine Arztpraxis (Queue) <i>Beispiel:</i> Simulation zur Auslastung einer Autowaschanlage und zur Untersuchung der Wartezeit der Kunden (Queue) <i>Beispiel:</i> Entwicklung eines Vokabeltrainers (List) <i>Beispiel:</i> Rangieren von Wagons auf einem Rangierbahnhof (Stack)</p>
<p>2. Lineare Liste (List) Feststellen der Unzulänglichkeiten der Klasse <code>Queue</code> für ein geplantes Projekt (z.B. Vokabeltrainer); Erarbeiten der Struktur einer linearen Liste (incl. graphischer Darstellung); Standardoperationen der linearen Liste; Entwickeln eines objektorientierten Designs des geplanten Projekts (Implementationsdiagramm mit Dokumentation der Methoden); Übernehmen der gegebenen Klasse <code>List</code> in die existierende</p>	

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>Klassenbibliothek; Entwickeln der Algorithmen und Implementieren der Methoden des Projekts unter Verwendung der Klassenbibliothek (insbes. der Klasse <code>List</code>); Aufgreifen der Such- und Sortierverfahren</p>	
<p>3. Stapel (Stack) Entwickeln der Anforderungen an eine Datenstruktur zur Lösung eines gegebenen Problems (z.B. Rangierproblem); Last-In-First-Out-Prinzip; Erarbeiten der Struktur eines Stapels (incl. graphischer Darstellung); Standardoperationen des Stapels; Implementieren der Klasse <code>Stack</code>; Übernehmen der Klasse <code>Stack</code> in die existierende Klassenbibliothek; Wrapper-Klassen und Autoboxing/Unboxing (zum Testen der Klasse <code>Stack</code>); Entwickeln von Algorithmen zur gegebenen Problemstellung; Verwenden der Klassenbibliothek (insbes. der Klasse <code>Stack</code>) zur Implementierung der erarbeiteten Algorithmen.</p>	

Q-V: Datenbanken**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Sicherheit

Inhalte	Beispiele, Ergänzungen, Hinweise
1. Aufbau und Funktionsweise eines Datenbanksystems Aufbau eines Datenbanksystems; Anforderungen an Datenbanksysteme; Simultane und sichere Nutzung eines Datenbanksystems	<i>Beispiel:</i> Schulverwaltung (Schüler belegen Kurse, die von Lehrern unterrichtet werden; Lehrer unterrichten (mehrere) Fächer; jedem Kurs ist mindestens ein Raum zugewiesen; ...)
2. Das Entity-Relationship-Modell Modellieren von Sachverhalten durch Entitäten und deren Attribute sowie durch Beziehungen zwischen Entitäten; Graphisches Darstellen des ER-Modells; Identifizieren einer Entität durch Primärschlüssel; Kardinalitäten von Beziehungen	<i>Beispiel:</i> Schulbuchverwaltung (Schüler leihen Bücher und besuchen eine Klasse, die von einem Lehrer geleitet wird) <i>Beispiel:</i> Rechnungen (Kunden erhalten Rechnungen, die Artikel enthalten)
3. Das relationale Modell und Normalisierung Grundprinzip des relationalen Modells; Überführen von Entitätsmengen in das relationale Modell; Überführen von Beziehungsmengen in das relationale Modell; Bestimmen des Primärschlüssels von Beziehungsrelationen; Berücksichtigen von Sonderfällen (z.B: Mehrfachattribute) bei der Überführung des ER-Modells in das relationale Modell; Optimieren des relationalen Modells durch die Zusammenfassung von Relationen (bei Beziehungen mit den Kardinalitäten 1:1 bzw. 1:n); Integrität des Datenbestandes (Datenintegrität und referenzielle Integrität); Redundanz, Anomalien und Inkonsistenz in einem Datenbestand;	<i>Beispiel:</i> Verwaltung eines Sportvereins (Mitglieder haben Konten bei einer Bank und nehmen an Übungsgruppen teil, jede Übungsgruppe wird von einem Mitglied geleitet, die Übungsgruppen trainieren an Trainingsorten, Hausmeister betreuen diese Trainingsorte) <i>Beispiel:</i> Personalinformationssystem (Mitarbeiter arbeiten an Projekten, die von jeweils einem Mitarbeiter geleitet werden)

Inhalte	Beispiele, Ergänzungen, Hinweise
Normalisieren von Relationen: 1. Normalform, funktionale Abhängigkeit von Attributen und 2. Normalform, transitive Abhängigkeit von Attributen und 3. Normalform	
4. Die Abfragesprache SQL „Einfache“ Projektion und Selektion; Vergleichsoperatoren; logische Operatoren; Umbenennen von Spaltennamen; Sortieren der Ausgabe; Entfernen von Duplikaten in der Ausgabe; kartesisches Produkt; Join (auch über mehr als 2 Tabellen); Vereinigung; Aggregatfunktionen; Gruppierungen; Formulieren von Bedingungen an Gruppen; Unterabfragen	

Q-VI: Sicherheit und Datenschutz in Netzstrukturen**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Einzelrechner und Rechnernetzwerke
- Nutzung von Informatiksystemen
- Sicherheit
- Wirkungen der Automatisierung

Inhalte	Beispiele, Ergänzungen, Hinweise
1. Klassifizieren von Computernetzwerken Klassifizieren nach der Ausdehnung (LAN, WAN); Klassifizieren nach der Architektur (Peer-to-Peer, Client-Server); Klassifizieren nach der Topologie (Bus-, Ring-, Stern-, vermaschte Topologie);	<i>Beispiel:</i> Rollenspiel zum Datenaustausch in Netzwerken, die nach bestimmten Topologien aufgebaut sind.
2. Netzwerkprotokolle und Schichtenmodelle Regelungen bzgl. der Kommunikation zwischen Computern in Form von Protokollen; Kommunikation in Netzwerken mithilfe von Schichten und Protokollen; Vorstellen des ISO-OSI-Schichtenmodells; Erarbeiten des TCP/IP-Schichtenmodells incl. der Aspekte: Segmentierung und paketbasierte Übermittlung, Anwendungsprozesse und Ports, IP-Adressen, Routing, Sicherungstechniken zur Erkennung von Fehlern bei der Datenübermittlung	<i>Beispiel:</i> Schriftliche Kommunikation zwischen zwei Managern einer Firma, die sich an unterschiedlichen Standorten befinden und verschiedene Sprachen sprechen (Schichtenmodell). <i>Beispiel:</i> Ablauf der Kommunikation zwischen zwei Rechnern beim Anfordern und Übermitteln einer Web-Seite. <i>Beispiel:</i> Vergleich der traditionellen Telefonie mit Voice-over-IP <i>Beispiel:</i> Rollenspiel zur Simulation der Arbeitsweise eines Routers
3. Verschlüsselungsverfahren Grundprinzipien der Kryptologie (Sender, Empfänger, Angreifer); Begriff des Substitutionsverfahrens; mono- und polyalphabetische Verschlüsselung; symmetrische und asymmetrische Verschlüsselung; Cäsar-, Vigenère- und RSA-Verschlüsselung jeweils mit den Aspekten: Ver- und Entschlüsselung, Sicherheit des Verfahrens	<i>Beispiel:</i> Ver- und Entschlüsselung von Botschaften mit den verschiedenen Verfahren <i>Falleispiel:</i> Erfassen und Auswerten von Kunden- und Ausleihdaten in einem VideoCenter. <i>Fallbeispiel:</i> Plagiat bei einer „wissenschaftlichen“ Arbeit
4. Fallbeispiele zu Datenschutzproblematik und Urheberrecht Volkszählungsurteil und Bundesdatenschutzgesetz; Untersuchen	

Inhalte	Beispiele, Ergänzungen, Hinweise
und Bewerten von Fallbeispielen bzgl. ethischer Fragen und unter den Aspekten: Datenschutz, Urheberrecht; Beispiele für Einschränkungen des Datenschutzes (Terrorabwehr); Datenschutz und Weiterentwicklung technischer Geräte	

Q-VII: Nichtlineare Strukturen (binäre Bäume)**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Ungeordneter Binärbaum (BinaryTree) Rekursive Definition des Binärbaums; Standardoperationen des Binärbaums; Traversieren von Binärbäumen (Preorder, Inorder, Postorder) incl. Implementierung mithilfe der Standardoperationen; Exkurs: Verarbeitung von Zeichenketten (Strings) in Java; Anwenden der Standardoperationen und der „Stringbearbeitung“ in einem Projekt (z.B. Morsekodierer); Implementieren der generischen Klasse <code>BinaryTree</code></p>	<p><i>Beispiel:</i> Morsebaum – Kodieren und Dekodieren von Klartext bzw. „Morsetext“ anhand eines Baums, in dem Zeichen gemäß dem Morsealphabet positioniert sind.</p> <p><i>Beispiel:</i> Huffman-Komprimierung von Text</p> <p><i>Beispiel:</i> Analyse der Zeichenhäufigkeiten in Texten</p>
<p>2. Anwendungsprojekt zum ungeordneten Binärbaum Analyse einer komplexeren Problemstellung; Nachvollziehen und „Durchspielen“ komplexer Algorithmen; Modellieren von Entwurfs- und Implementationsdiagrammen; Entwickeln „grober“ Strategien zur Umsetzung geplanter Java-Methoden; Analyse gegebener Quelltextpassagen; Entwurf und Implementierung von Algorithmen in Java</p>	
<p>3. Geordneter Binärbaum (BinarySearchTree) Definition des geordneten Binärbaums; Suchen von Datenobjekten sowie Einfügen und Löschen von Datenobjekten und Knoten in binären Suchbäumen; Notwendigkeit einer Ordnungsrelation in</p>	

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>binären Suchbäumen; Gewährleisten der Existenz einer Ordnungsrelation von Datenobjekten, die in Suchbäumen verwaltet werden sollen, mithilfe eines geeigneten generischen Interface (<code>ComparableContent<ContentType></code>); Standardoperationen eines binären Suchbaums; Verwenden der Standardoperationen in einem Anwendungsprojekt (incl. Entwurf der entsprechenden Entwurfs- und Implementationsdiagramme)</p>	

Q-VIII: Formale Sprachen (Automaten und Grammatiken)

Zentrale Kompetenzen:

- Argumentieren
- Modellieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

- Endliche Automaten
- Grammatiken regulärer Sprachen
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

Inhalte	Beispiele, Ergänzungen, Hinweise
<p>1. Natürliche und formale Sprachen Abgrenzung zwischen natürlichen und formalen Sprachen; anwendungsbezogenes Beispiel für eine formale Sprache; Syntax und Semantik formaler Sprachen</p>	<p><i>Beispiel:</i> Formale Sprache zur „Kommunikation“ mit einem Herd</p> <p><i>Beispiel:</i> Grammatik zur Beschreibung der U- und S-Bahnlinien einer Kleinstadt</p>
<p>2. Reguläre Grammatiken Definition des Begriffs Grammatik; Beispiele für Grammatiken (z.B. mit rekursiven Produktionen); reguläre Grammatiken und reguläre Sprachen</p>	<p><i>Beispiele:</i> Formale (reguläre) Sprachen, deren Wörter aus den Zeichen a und b (oder 0 und 1) bestehen und die bestimmte vorgegebene Eigenschaften erfüllen</p>
<p>3. Endliche Automaten Einführung deterministischer Akzeptoren; Beispiele für deterministische Akzeptoren (z.B. mit absorbierenden Zuständen); Zusammenhang zwischen deterministischen Akzeptoren und regulären Grammatiken; Umwandeln von deterministischen Akzeptoren in reguläre Grammatiken und umgekehrt; Abgrenzung deterministischer und nichtdeterministischer Akzeptoren</p>	<p><i>Beispiele:</i> Formale Sprachen die die natürlichen Zahlen bzw. die ganzen Zahlen beschreiben</p> <p><i>Beispiel:</i> Formale Sprachen, die den Aufbau eines gültigen Computerpasswords, das bestimmte Bedingungen erfüllen muss, beschreiben</p>
<p>4. Grenzen regulärer Sprachen Beispiel für eine nicht reguläre Sprache und entsprechende Begründung (bzw. Beweis); ggf. kurze Einführung von Kellerautomaten und kontextfreien Grammatiken sowie Konstruktion eines Kellerautomaten, der die gegebene nicht reguläre aber kontextfreie Sprache akzeptiert;</p>	<p><i>Beispiel:</i> Endlicher Automat, der prüft, ob bei gegebenen Noten die Versetzung erreicht wird</p> <p><i>Beispiel:</i> Sprache $L = \{ a^n b^n \mid n \in \mathbb{N} \}$ als Beispiel einer nicht regulären Sprache</p>

Q-IX: Funktionsweise eines Computers und Grenzen der Automatisierbarkeit**Zentrale Kompetenzen:**

- Argumentieren

Inhaltsfelder:

- Algorithmen
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Analyse, Entwurf und Implementierung von Algorithmen
- Einzelrechner und Rechnernetzwerke
- Grenzen der Automatisierung

Inhalte	Beispiele, Ergänzungen, Hinweise
1. von-Neumann-Architektur Grundkomponenten eines von-Neumann-Rechners; Interner Aufbau der Grundkomponenten (z.B. diverse Register, ALU, ...); von-Neumann'scher Flaschenhals; Befehlsarten für das von-Neumann-Modell; Befehlszyklus; Rechnermodell mit beispielhafter Maschinensprache; Analyse von Maschinenprogrammen dieser Maschinensprache	<i>Beispiel:</i> Maschinenprogramm zur Multiplikation zweier Zahlen durch wiederholte Addition <i>Beispiel:</i> Maschinenprogramm zur Ermittlung der Häufigkeitsverteilung von Zahlen <i>Beispiel:</i> Paradoxon von Zenon von Elea (Achill und die Schildkröte)
2. Grenzen der Automatisierbarkeit Nicht terminierende Algorithmen; Halteproblem; Untersuchen der Lösbarkeit des Halteproblems; berechenbare und nicht berechenbare Probleme; Beurteilen des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Grenzen	<i>Beispiel:</i> Collatz-Funktion

4. Zuordnung der konkretisierten Kompetenzerwartungen bis zum Ende der Qualifikationsphase zu den Unterrichtsvorhaben

Kompetenzerwartungen	Unterrichtsvorhaben								
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX
Inhaltsfeld 1: Daten und ihre Strukturierung									
<u>Inhaltliche Schwerpunkte:</u> - Objekte und Klassen - Datenbanken									
<i>Objekte und Klassen</i> Die Schülerinnen und Schüler									
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),				X			X		
- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),		X		X			X		
- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),	X			X			X		
- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),	X						X		
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),	X	X	X	X			X		
- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M),	X								
- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),	X	X	X	X			X		
- stellen die Kommunikation zwischen Objekten grafisch dar (D),				X					

Kompetenzerwartungen	Unterrichtsvorhaben								
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX
- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),	x			x			x		
- dokumentieren Klassen (D),				x			x		
- analysieren und erläutern objektorientierte Modellierungen (A),	x			x			x		
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).	x	x		x			x		
<i>Datenbanken</i> Die Schülerinnen und Schüler									
- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),					x				
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D),					x				
- modifizieren eine Datenbankmodellierung (M),					x				
- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),					x				
- bestimmen Primär- und Sekundärschlüssel (M),					x				
- analysieren und erläutern eine Datenbankmodellierung (A),					x				
- erläutern die Eigenschaften normalisierter Datenbankschemata (A),					x				
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),					x				
- überführen Datenbankschemata in die 1. bis 3. Normalform (M).					x				

Kompetenzerwartungen	Unterrichtsvorhaben																	
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX									
Inhaltsfeld 2: Algorithmen																		
<u>Inhaltliche Schwerpunkte:</u>																		
- Analyse, Entwurf und Implementierung von Algorithmen																		
- Algorithmen in ausgewählten informatischen Kontexten																		
<i>Analyse, Entwurf und Implementierung von Algorithmen</i>																		
Die Schülerinnen und Schüler																		
- analysieren und erläutern Algorithmen und Programme (A),										x	x	x	x			x		x
- modifizieren Algorithmen und Programme (I),										x	x	x	x			x		
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),											x	x	x			x		
- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),											x	x	x			x		
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),		x	x	x			x											
- testen Programme systematisch anhand von Beispielen (I).	x	x	x	x			x											
<i>Algorithmen in ausgewählten informatischen Kontexten</i>																		
Die Schülerinnen und Schüler																		
- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),													x			x		
- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),												x	x			x		
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),											x	x						
- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D).					x													

Kompetenzerwartungen	Unterrichtsvorhaben																	
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX									
Inhaltsfeld 3: Formale Sprachen und Automaten																		
<u>Inhaltliche Schwerpunkte:</u> - Syntax und Semantik einer Programmiersprache - Endliche Automaten - Grammatiken regulärer Sprachen - Möglichkeiten und Grenzen von Automaten und formalen Sprachen																		
<i>Syntax und Semantik einer Programmiersprache</i> Die Schülerinnen und Schüler																		
- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),										x	x	x	x			x		
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),										x	x	x	x			x		
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),										x	x	x	x			x		
- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),														x				
- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I).														x				
<i>Endliche Automaten</i> Die Schülerinnen und Schüler																		
- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A),																	x	
- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D),								x										
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),								x										
- stellen endliche Automaten in Tabellen oder Graphen dar und								x										

Kompetenzerwartungen	Unterrichtsvorhaben								
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX
überführen sie in die jeweils andere Darstellungsform (D),									
- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M).								x	
<i>Grammatiken regulärer Sprachen</i> Die Schülerinnen und Schüler									
- analysieren und erläutern Grammatiken regulärer Sprachen (A),								x	
- modifizieren Grammatiken regulärer Sprachen (M),								x	
- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),								x	
- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),								x	
- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M),								x	
- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).								x	
<i>Möglichkeiten und Grenzen von Automaten und formalen Sprachen</i> Die Schülerinnen und Schüler									
- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A).								x	

Kompetenzerwartungen	Unterrichtsvorhaben								
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX
Inhaltsfeld 4: Informatiksysteme									
<u>Inhaltliche Schwerpunkte:</u> - Einzelrechner und Rechnernetzwerke - Nutzung von Informatiksystemen - Sicherheit									
<i>Einzelrechner und Rechnernetzwerke</i> Die Schülerinnen und Schüler									
- erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),									X
- beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A).						X			
<i>Nutzung von Informatiksystemen</i> Die Schülerinnen und Schüler									
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),	X	X	X	X	X	X	X		
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),	X	X	X	X	X		X		
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I).	X	X	X	X	X		X		
<i>Sicherheit</i> Die Schülerinnen und Schüler									
- erläutern Eigenschaften, Funktionsweisen und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),					X				
- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).						X			

Kompetenzerwartungen	Unterrichtsvorhaben								
	Q-I	Q-II	Q-III	Q-IV	Q-V	Q-VI	Q-VII	Q-VIII	Q-IX
Inhaltsfeld 5: Informatik, Mensch und Gesellschaft									
Inhaltliche Schwerpunkte: - Wirkungen der Automatisierung - Grenzen der Automatisierung									
<i>Wirkungen der Automatisierung</i> Die Schülerinnen und Schüler									
- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),						x			
- 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).						x			
<i>Grenzen der Automatisierung</i> Die Schülerinnen und Schüler									
- untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).									x

5. Grundsätze der Leistungsbewertung

Hinweis: Zusätzlich zu den nachfolgenden Festlegungen können ergänzend weitere Instrumente zur Leistungsüberprüfung zum Einsatz kommen.

Beurteilungsbereich „Klausuren“

Anzahl und Dauer der Klausuren in den einzelnen Halbjahren der gymnasialen Oberstufe

- Einführungsphase:
 1. Halbjahr: eine Klausur, 2 Unterrichtsstunden
 2. Halbjahr: zwei Klausuren, jeweils 2 Unterrichtsstunden
- Qualifikationsphase Q1:
 1. Halbjahr: zwei Klausuren, jeweils 2 Unterrichtsstunden
 2. Halbjahr: zwei Klausuren, jeweils 2 Unterrichtsstunden
- Qualifikationsphase Q2:
 1. Halbjahr: zwei Klausuren, jeweils 3 Unterrichtsstunden
 2. Halbjahr: eine Klausur („Vorabiklausur“), 3 Zeitstunden, Abiturklausur: 3 Zeitstunden

Anstelle einer Klausur kann nach Maßgabe der Schule im zweiten Halbjahr der Qualifikationsphase Q1 eine Facharbeit angefertigt werden.

Aufgabentypen und Anforderungsbereiche in Klausuren

Die Aufgabentypen in Klausuren orientieren sich an den im Kernlehrplan (Kapitel 3) genannten Überprüfungsformen. Weitere über die Auflistung im Kernlehrplan hinausgehende Überprüfungsformen sind möglich.

Die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 4 des Kernlehrplans zu beachten.

Verwenden von Operatoren in Aufgabenstellungen bei Klausuren

Bei der Formulierung der Aufgabenstellungen in Klausuren werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik berücksichtigt. Diese fachspezifischen Operatoren werden zuvor eingeführt und erläutert.

Bewertung von Klausuren

Die Bewertung der Klausuren erfolgt jeweils auf Basis eines Bewertungsrasters mit Hilfspunkten. Die Zuordnung der erreichten Punktzahl zu den Notenstufen orientiert sich am Zuordnungsschema für die schriftliche Abiturprüfung des Zentralabiturs. Von diesem kann aber, falls besondere Umstände vorliegen, begründet abgewichen werden. In der Regel soll die Note ausreichend (5 Punkte) bei Erreichen von 45% der Hilfspunkte erteilt werden.

Gemäß Kapitel 3 des Kernlehrplans und §13 (2) der APO-GOST ist bei Klausuren die Darstellung bei der Leistungsbewertung hinreichend zu berücksichtigen. Eine schwache Darstellung kann also ebenso wie „Verstöße gegen die sprachliche Richtigkeit in der deutschen Sprache“ zu einer Absenkung der Leistungsbewertung führen.

Beurteilungsbereich „Sonstige Leistungen“

Die Kriterien zum Beurteilungsbereich „sonstige Leistungen“ werden zu Beginn eines jeden Schuljahres genannt. Individuelle Leistungsrückmeldungen erfolgen jeweils zum Quartalsende, bei auffälligen Leistungsveränderungen, auf Anfrage des Schülers und zu Sprechtagen.

Leistungsaspekte

Mündliche Leistungen:

- Beteiligung am Unterrichtsgespräch
- Erläuterung von Algorithmen bzw. von Quelltexten
- Mitarbeit in Partner-/Gruppenarbeitsphasen
- Präsentation von Arbeitsergebnissen
- Zusammenfassung zur Vor- und Nachbereitung des Unterrichts
- ggf. Anfertigung und Präsentation von Referaten
- ggf. Lernerfolgsüberprüfung durch ein Fachgespräch zwischen dem Schüler und der Lehrkraft

Praktische Leistungen am Computer:

- Implementierung, Test und Anwendung von Informatiksystemen

Sonstige schriftliche Leistungen:

- Bearbeitung von schriftlichen Aufgaben im Unterricht
- Dokumentation von Arbeitsprozessen und Quelltexten
- ggf. Anfertigung von Stundenprotokollen
- ggf. Lernerfolgsüberprüfung durch kurze schriftliche Übungen, die sich auf den Stoff der letzten ca. 4-6 Unterrichtsstunden bzw. auf die Hausaufgaben beziehen
- ggf. Arbeitsmappe

Bearbeitung von Hausaufgaben:

- regelmäßige und gewissenhafte Bearbeitung von Hausaufgaben
Es kann von den Schülerinnen und Schülern verlangt werden, Hausaufgaben per E-Mail bis zu einem festgelegten Termin an den Fachlehrer zu schicken. Dies gilt insbesondere für erstellte Softwareprodukte.

Bewertung der sonstigen Leistungen

Die folgenden Bewertungskriterien gelten für die mündlichen, die schriftlichen und die praktischen Formen der sonstigen Mitarbeit.

Die Bewertung stützt sich auf

- die Qualität der Beiträge
- die Quantität der Beiträge
- die Kontinuität der Beiträge

Besonderes Augenmerk wird dabei auf folgende Aspekte gelegt:

die sachliche Richtigkeit; die Ziel- und Ergebnisorientierung; die Präzision; die Verständlichkeit; die Darstellungskompetenz; die angemessene Verwendung der Fachsprache; die Selbstständigkeit im Arbeitsprozess

Bei Gruppenarbeiten auch auf:

das Einbringen in die Arbeit der Gruppe; die Übernahme von Verantwortung; die Qualität des entwickelten Produkts

Bei praktischen Arbeiten darüber hinaus auf:

die Qualität des entwickelten Programms (incl. Dokumentation); die Übersichtlichkeit, Lesbarkeit und Verständlichkeit des Quelltextes; die Bereitschaft und die Geduld, syntaktische und semantische Fehler im Quelltext aufzuspüren und zu beseitigen; den Grad an Selbstständigkeit; die Aufnahme von Beratung durch die Lehrkraft