

Einführungsphase E (ca. 70 60-Minuten-Stunden)	
<b>E: Informationstechnische Grundbildung (11 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<b>Tabellenkalkulation (11 h)</b> <ul style="list-style-type: none"> <li>• Relative und absolute Adressierung</li> <li>• Datentypen</li> <li>• Formeln</li> <li>• (Bedingte) Formatierung</li> <li>• Wenn-Funktion</li> <li>• Visualisierung mit Diagrammen</li> </ul>	<ul style="list-style-type: none"> <li>• erfassen, erkunden und lösen Probleme</li> <li>• arbeiten mit Daten und deren Beziehungen, beschreiben Veränderungen und bestimmen mit Hilfe des PCs Ergebnisse</li> <li>• lösen mathematische Probleme mit dem Computer als Werkzeug</li> <li>• erkennen abstrakter Zusammenhänge</li> <li>• modellieren und stellen mit Hilfe des PCs dar</li> </ul>

<b>E: Was macht Informatik? – Einführung in die Inhaltsfelder der Informatik (12 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<b>Informatiksysteme und ihr genereller Aufbau (2 h)</b> <ul style="list-style-type: none"> <li>• Daten und ihre Strukturierung</li> <li>• Algorithmen</li> <li>• Informatiksysteme</li> <li>• Informatik, Mensch und Gesellschaft</li> </ul>	<ul style="list-style-type: none"> <li>• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen</li> <li>• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst</li> </ul>
<b>Identifikation von Objekten und Klassen (2 h)</b> <ul style="list-style-type: none"> <li>• An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt.</li> <li>• Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt.</li> <li>• Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.</li> </ul> <b>Analyse von Objekten und Klassen im Greenfoot-Szenario (3 h)</b> <ul style="list-style-type: none"> <li>• Schritte der objektorientierten Analyse, Modellierung und Implementation</li> <li>• Analyse und Erprobung der Objekte im <b>Greenfoot-Szenario (optional)</b></li> </ul> <b>Implementierung einfacher Aktionen in Greenfoot (5 h)</b> <ul style="list-style-type: none"> <li>• Quelltext einer Java-Klasse</li> <li>• Implementation eigener Methoden,</li> <li>• Optional: Dokumentation JavaDoc</li> <li>• Programme übersetzen (Aufgabe des Compilers) und testen</li> </ul>	<ul style="list-style-type: none"> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen,</li> <li>• stellen den Zustand eines Objekts dar,</li> <li>• modellieren Klassen mit ihren Attributen und ihren Methoden,</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache,</li> <li>• implementieren Klassen in einer Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken</li> </ul>

<b>E: Algorithmische Grundstrukturen in Java (16 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Algorithmen</b></p> <ul style="list-style-type: none"> <li>• Wiederholungen (While-Schleife, Zählschleife)</li> <li>• bedingte Anweisungen</li> <li>• Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT</li> <li>• Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme</li> </ul> <p><b>Variablen und Methoden</b></p> <ul style="list-style-type: none"> <li>• Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife</li> <li>• Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert</li> <li>• Implementierung von Konstruktoren</li> <li>• Realisierung von Attributen</li> </ul>	<ul style="list-style-type: none"> <li>• analysieren und erläutern einfache Algorithmen und Programme,</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar,</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen zu,</li> <li>• modifizieren einfache Algorithmen und Programme,</li> <li>• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen,</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken,</li> <li>• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache,</li> <li>• testen Programme schrittweise anhand von Beispielen,</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode Methoden</li> </ul>

<b>E: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele (18 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Umsetzung von Anforderungen in Entwurfsdiagramme (3 h)</b></p> <ul style="list-style-type: none"> <li>• Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert</li> <li>• Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert.</li> </ul> <p><b>Implementationsdiagramme als erster Schritt der Programmierung (5 h)</b></p> <ul style="list-style-type: none"> <li>• Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</li> <li>• Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</li> <li>• Entwicklung von Klassendokumentationen</li> <li>• Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung</li> </ul> <p><b>Vererbungsbeziehungen (10 h)</b></p> <ul style="list-style-type: none"> <li>• Das Grundprinzip der Vererbung wird erarbeitet.</li> <li>• Die Vorteile der Vererbungsbeziehungen</li> <li>• Vererbung wird implementiert</li> </ul>	<ul style="list-style-type: none"> <li>• analysieren und erläutern eine objektorientierte Modellierung,</li> <li>• stellen die Kommunikation zwischen Objekten grafisch dar,</li> <li>• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen,</li> <li>• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen,</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu,</li> <li>• ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu,</li> <li>• modellieren Klassen unter Verwendung von Vererbung,</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken,</li> <li>• testen Programme schrittweise anhand von Beispielen,</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode,</li> <li>• analysieren und erläutern einfache Algorithmen und Programme,</li> <li>• modifizieren einfache Algorithmen und Programme,</li> <li>• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar,</li> <li>• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar,</li> <li>• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden</li> </ul>

<b>E: Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung (12 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Analoge und digitale Aufbereitung und Verarbeitung von Daten (2 h)</b></p> <ul style="list-style-type: none"> <li>• Erarbeitung der Unterschiede von analog und digital</li> <li>• Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital</li> </ul> <p><b>Der Umgang mit binärer Codierung von Informationen (3 h)</b></p> <ul style="list-style-type: none"> <li>• Das binäre (und hexadezimale) Zahlensystem</li> <li>• Binäre Informationsspeicherung</li> <li>• Binäre Verschlüsselung</li> </ul> <p><b>Aufbau informatischer Systeme (7 h)</b></p> <ul style="list-style-type: none"> <li>• Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen</li> <li>• Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips</li> </ul>	<ul style="list-style-type: none"> <li>• bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen</li> <li>• stellen ganze Zahlen und Zeichen in Binärcodes dar</li> <li>• interpretieren Binärcodes als Zahlen und Zeichen</li> <li>• beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“</li> <li>• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation</li> <li>• simulieren die CPU, das Rechen- und Steuerwerks mit Johnny (Johnny: optional)</li> </ul>

Qualifikationsphase Q <sub>1</sub> (ca. 70 60-Minuten-Stunden).	Hinweis. Die Themen in Q <sub>1</sub> und Q <sub>2</sub> können getauscht werden.
<b>Q<sub>1</sub>: Datenstrukturen in Java (10 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<b>Lineare Datenstrukturen (10 h)</b> <ul style="list-style-type: none"> <li>• Begriff Datenstruktur Anforderungen an eine Datenstruktur (DS)</li> <li>• Unterschied statische und dynamische DS</li> <li>• Statische DS Array <ul style="list-style-type: none"> <li>○ Eigenschaften und Standardoperationen für ein- und zweidimensionale Arrays</li> <li>○ Modellierung und Implementierung von Anwendungen</li> </ul> </li> <li>• Dynamische DS → Queue, Stack, List <ul style="list-style-type: none"> <li>○ Eigenschaften und Standardoperationen und Methoden der Datenstrukturen</li> <li>○ Erläuterung von Problemstellungen, die nach dem FIFO- bzw. LIFO-Prinzip bearbeitet werden</li> <li>○ Modellierung und Implementierung von Anwendungen unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• erläutern Operationen dynamischer linearer Datenstrukturen</li> <li>• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen</li> <li>• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode</li> <li>• stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau</li> <li>• modellieren Klassen und dokumentieren Klassen</li> <li>• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken</li> <li>• verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie</li> <li>• untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts</li> </ul>

<b>Q1: Algorithmische Grundstrukturen in Java (15 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Algorithmen (3 h)</b></p> <ul style="list-style-type: none"> <li>• Eigenschaften von Algorithmen</li> <li>• Strukturieren von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</li> <li>• Rekursion und Iteration; Analyse und Entwicklung von rekursiven Algorithmen</li> </ul> <p><b>Iteratives und rekursives Suchen und Sortieren (10 h)</b></p> <ul style="list-style-type: none"> <li>• Lineare Suche, Binäre Suche, Hashing</li> <li>• Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Insertion Sort)</li> <li>• Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort)</li> <li>• Untersuchung der Verfahren bzgl. Laufzeit und Speicherplatzbedarf</li> <li>• Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)</li> </ul> <p><b>Verantwortung der Informatik (2 h)</b></p> <ul style="list-style-type: none"> <li>• Informatiksysteme aus Produzent- und Anwendersicht</li> <li>• Informatik und Ethik</li> </ul>	<ul style="list-style-type: none"> <li>• analysieren, erläutern und modifizieren Algorithmen und Programme</li> <li>• stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar</li> <li>• entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“</li> <li>• implementieren iterative und rekursive Algorithmen unter Verwendung von dynamischen Datenstrukturen</li> <li>• testen Programme systematisch anhand von Beispielen</li> <li>• implementieren und erläutern iterative und rekursive Such- und Sortierverfahren</li> <li>• beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen</li> <li>• beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen</li> <li>• nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen</li> <li>• interpretieren Fehlermeldungen und korrigieren den Quellcode</li> <li>• untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts</li> <li>• untersuchen und bewerten von Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte</li> </ul>

<b>Q<sub>1</sub>: Kommunikation in Netzwerken (12 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Menschen kommunizieren – ohne und mit Technik (1 - 2 h)</b></p> <ul style="list-style-type: none"> <li>• Wie läuft eine Kommunikation zwischen (2 oder mehr) Menschen ohne Technik ab? (Shannon-Weaver-Modell)</li> <li>• Anforderungen an technische Kommunikationen für eine fehlerfreie Übertragung</li> <li>• Geschichtliche Entwicklung</li> </ul>	<ul style="list-style-type: none"> <li>• stellen Abläufe und Bedingungen von erfolgreichen Kommunikationsprozesse dar</li> <li>• übertragen diese Erkenntnisse auf die technische Kommunikation</li> <li>• lernen alte technische Kommunikationsformen kennen</li> </ul>
<p><b>Ohne Protokolle läuft nichts – Netzwerke (10 h)</b></p> <ul style="list-style-type: none"> <li>• Bestandteile eines Netzwerkes und deren Funktion</li> <li>• Verschiedene Topologien</li> <li>• Protokolle – Kommunikationsregeln</li> <li>• OSI-Schichtenmodell</li> <li>• TCP/IP Modell (Anwendungsschicht, Transportschicht, Internetschicht; optional: Netzzugangsschicht)</li> <li>• IPv4-Struktur (Netz- und Hostteil), Subnetting, optional: Klassen A, B, C</li> <li>• Simulation von Netzwerken mit Filius</li> </ul>	<ul style="list-style-type: none"> <li>• erforschen das eigene (Schul-)Netz</li> <li>• skizzieren die Topologie an unserer Schule</li> <li>• lernen die Schichten des OSI-Modells (TCP/IP-Modells) kennen</li> <li>• kennen den Unterschied zwischen den Netzwerkklassen</li> </ul>



<b>Q<sub>1</sub>: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten (33 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Nutzung von relationalen Datenbanken (16h)</b></p> <ul style="list-style-type: none"> <li>• Aufbau von Datenbanksystemen und Grundbegriffe <ul style="list-style-type: none"> <li>○ Aufgaben und Eigenschaften eines Datenbanksystems</li> <li>○ Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema</li> <li>○ Problematisierung von Redundanzen, Anomalien und Inkonsistenzen</li> </ul> </li> <li>• SQL-Abfragen <ul style="list-style-type: none"> <li>○ Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN)</li> <li>○ Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke)</li> </ul> </li> <li>• Vertiefung an einem weiteren Datenbankbeispiel</li> </ul>	<ul style="list-style-type: none"> <li>• erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung,</li> <li>• analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage,</li> <li>• verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren,</li> <li>• ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen,</li> </ul>

<b>Q<sub>1</sub>: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten (35 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Modellierung von relationalen Datenbanken (17 h)</b></p> <ul style="list-style-type: none"> <li>• Datenbankentwurf durch ER-Diagramme <ul style="list-style-type: none"> <li>○ Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>○ Erläuterung und Erweiterung einer Datenbankmodellierung</li> </ul> </li> <li>• Entwicklung eines relationalen Modells aus einem Datenbankentwurf <ul style="list-style-type: none"> <li>○ Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln</li> </ul> </li> <li>• Normalformen <ul style="list-style-type: none"> <li>○ Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul> </li> </ul> <p><b>Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b></p>	<ul style="list-style-type: none"> <li>• ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten,</li> <li>• stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar</li> <li>• modifizieren eine Datenbankmodellierung,</li> <li>• modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema,</li> <li>• bestimmen Primär- und Sekundärschlüssel,</li> <li>• analysieren und erläutern eine Datenbankmodellierung,</li> <li>• erläutern die Eigenschaften normalisierter Datenbankschemata,</li> <li>• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften.</li> <li>• überführen Datenbankschemata in die 1. bis 3. Normalform</li> </ul>

<b>Q<sub>1</sub>: Optional: Datensicherheit und Datenschutz (3 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<b>Datenschutz und Datensicherheit (3 h)</b>	<ul style="list-style-type: none"> <li>• nehmen das Recht auf informationelle Selbstbestimmung wahr und wenden die Gesetze zum Datenschutz an</li> <li>• analysieren und erläutern: Sicherheit im Netz, Schutz lokaler Netze vor Angriffen von außen, analysieren politische und gesellschaftliche Rahmenbedingungen wichtiger informatischer Entwicklungen und beurteilen deren Wirkung</li> </ul>

Qualifikationsphase Q <sub>2</sub> (ca. 56 60-Minuten-Stunden)	Hinweis. Die Themen in Q <sub>1</sub> und Q <sub>2</sub> können getauscht werden.
<b>Q<sub>2</sub>: Kryptologie und Datensicherheit (20 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<b>Das Caesar-Verfahren (4 h)</b> <ul style="list-style-type: none"> <li>• Grundbegriffe der Kryptologie (Ver- und Entschlüsselung, Geheimtextalphabet, Klartextalphabet, ...)</li> <li>• Geschichte der Kryptologie</li> <li>• Modulare Addition und Subtraktion</li> <li>• Optional: Kryptoanalyse des Caesar-Verfahrens</li> <li>• Optional: weitere symmetrische Verschlüsselungsverfahren</li> <li>• Symmetrische und asymmetrische Verschlüsselung</li> </ul>	<ul style="list-style-type: none"> <li>• kennen und wenden das Caesar-Verfahren als typische monoalphabetische Verschlüsselung an</li> <li>• brechen den Caesar-Code durch Ausschöpfen des Schlüsselraums</li> <li>• können symmetrische und asymmetrische Verschlüsselungsverfahren unterscheiden</li> </ul>
<b>Multiplikative Verschlüsselung (5 h)</b> <ul style="list-style-type: none"> <li>• Optional: Ver- und Entschlüsselung durch Modulare Multiplikation</li> <li>• Optional: Anwendung des Verschlüsselungsverfahrens auf große Zahlen (Blockchiffre)</li> <li>• Einführung des Prinzips der asymmetrischen Verschlüsselung (Problem der Schlüsselverteilung und der authentischen Kommunikation)</li> <li>• Erweiterter euklidischer Algorithmus zur Bestimmung der modularen Inversen</li> </ul>	<ul style="list-style-type: none"> <li>• optional: führen die multiplikative Verschlüsselung (auch mit großen Zahlen) durchführen können</li> <li>• optional: werden mit dem Verfahren der Blockchiffrierung vertraut</li> <li>• wenden die Grundprinzipien der asymmetrischen Kryptographie an</li> <li>• optional: beurteilen die mangelnde Eignung der multiplikativen Verschlüsselung für die asymmetrische Kryptographie</li> </ul>
<b>Das RSA-Verfahren (6 h)</b> <ul style="list-style-type: none"> <li>• Chiffrierung durch modulares Potenzieren</li> <li>• Schlüsselgenerierung beim RSA-Verfahren</li> <li>• Einwegfunktionen</li> <li>• Nachweis der Korrektheit des RSA-Algorithmus</li> </ul>	<ul style="list-style-type: none"> <li>• wenden die Ver- und Entschlüsselung nach dem RSA-Verfahren an</li> <li>• verstehen das Prinzip zur Bestimmung von (großen) Primzahlen</li> <li>• erfassen die Einweg-Eigenschaft der Multiplikation großer Primzahlen erfassen</li> <li>• generieren Schlüsselpaare</li> <li>• beurteilen die Sicherheit des RSA-Verfahrens und vollziehen die Korrektheit des RSA-Verfahrens</li> </ul>

<p><b>Anwendungen des RSA-Verfahrens (5 h)</b></p> <ul style="list-style-type: none"><li>• Anwendungen und Probleme des RSA-Verfahrens</li><li>• Zentrale Zertifizierung vs. „Web of Trust“</li><li>• Schutz der Privatsphäre vs. Schutz vor Kriminellen und Terroristen</li></ul>	<ul style="list-style-type: none"><li>• verstehen das Prinzip der digitalen Unterschrift</li><li>• wenden ein Verschlüsselungssystem, das RSA verwendet (PGP o. ä.) an</li><li>• entwickeln ein Problembewusstsein für die Schlüssel-zertifizierung</li><li>• wägen Interessenkonflikte bei der starken Verschlüsselung ab</li></ul>
--	--

<b>Q2: Formale Sprachen und Automaten (21 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<p><b>Grammatiken und reguläre Sprachen (6 h)</b></p> <ul style="list-style-type: none"> <li>• Begriffe: Syntax, Semantik, Pragmatik, Wörter über einem Alphabet</li> <li>• Erarbeitung der formalen Darstellung regulärer Grammatiken und Syntaxdiagramme; EBNF</li> <li>• Untersuchung, Modifikation und Entwicklung von Grammatiken <ul style="list-style-type: none"> <li>○ Sprachbeschreibung mit Grammatiken</li> <li>○ Sprachbeschreibung mit regulären Ausdrücken</li> </ul> </li> </ul> <p><b>Endliche Automaten (10 h)</b></p> <ul style="list-style-type: none"> <li>• Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</li> <li>• Untersuchung, Darstellung und Entwicklung endlicher Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</li> <li>• Entwicklung regulärer Grammatiken zu endlichen Automaten</li> <li>• Spracherkennung mit InES und InGE</li> <li>• Grenzen endlicher Automaten</li> </ul> <p><b>Kellerautomaten (5 h)</b></p> <ul style="list-style-type: none"> <li>• deterministische und nichtdeterministische Kellerautomaten</li> <li>• Spracherkennung mit Kellerautomaten</li> <li>• Kontextfreie Sprachen</li> </ul> <p><b>Turingmaschine (optional)</b></p> <ul style="list-style-type: none"> <li>• Spracherkennung mit Turingmaschinen</li> </ul>	<ul style="list-style-type: none"> <li>• unterscheiden Syntax und Semantik</li> <li>• beschreiben formale Sprachen und stellen diese dar</li> <li>• analysieren, erläutern und definieren Grammatiken von Sprachen</li> <li>• ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird</li> <li>• entwickeln, modifizieren und implementieren erkennende Automaten zur Syntaxprüfung</li> <li>• analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben</li> <li>• entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik</li> <li>• entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten</li> <li>• modifizieren Grammatiken regulärer Sprachen</li> <li>• entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt</li> <li>• stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform</li> <li>• ermitteln die Sprache, die ein endlicher Automat akzeptiert</li> <li>• beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken</li> <li>• erkennen prinzipiell Grenzen des Computereinsatzes und zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf</li> </ul>

<b>Q<sub>2</sub>: Systementwicklung (15 h)</b>	
<b>Inhaltliche Schwerpunkte</b>	<b>Kompetenzen</b>
	Die SuS
<b>Projektdurchführung</b> <ul style="list-style-type: none"> <li>• abhängig vom Thema können auch Sachgebiete der anderen Jahrgangsstufen berührt werden</li> <li>• Themenwahl der Projektarbeit orientiert sich an den Interessen der Schülerinnen und Schüler</li> <li>• Auswahl: <ul style="list-style-type: none"> <li>○ DB-Entwurfsverfahren</li> <li>○ Elemente des Software-Engineering (z.B. systematisches Testen, Programmdokumentation)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• vertiefen und erweitern die in den Jahrgangsstufen E und Q1 erarbeiteten Grundlagen</li> <li>• wenden die grundsätzlichen Prinzipien der Projektarbeit an</li> <li>• planen von Projekten</li> <li>• arbeiten im Team Dokumentation der Projektarbeit</li> </ul>