Klasse 11			**Informatik 2-stündig**
Einführung in die objektorientierte Modellierung und Programmierung			
20	**Leitidee 3: Problemlösen und Modellieren** Die Schülerinnen und Schüler • kennen ein Konzept der objektorientierten Modellierung; • können Beziehungen zwischen Klassen und die Kommunikation zwischen Objekten analysieren und beschreiben; • können ein Modell entwerfen und analysieren. **Leitidee 2: Algorithmen und Daten** Die Schülerinnen und Schüler können • Benutzerschnittstellen mit einfachen Komponenten gestalten; • Techniken zur Modularisierung einsetzen. **Leitidee 3: Problemlösen und Modellieren** Die Schülerinnen und Schüler • kennen grundlegende Prinzipien beim Problemlösen und diese anwenden; • können ein Problem arbeitsteilig im Team lösen; • können den Problemlöseprozess strukturieren; • können eine Lösung dokumentieren, präsentieren und vertreten.	OOM Top-Down-Entwurf Objekt, Klasse, Attribut, Methode einfache Datentypen, Variablenkonzept Klassendiagramme Geheimnisprinzip Zustand und Verhalten eines Objektes	Einsatz einer geeigneten Entwicklungsumgebung z.B. BlueJ oder JavaEditor Beginn mit einer einzelnen Klasse noch ohne Aggregation, Assoziation Hier werden die oben formulierten Methoden ausprogrammiert MVC-Modell

Algorithmen			
35	**Leitidee 2: Algorithmen und Daten** Die Schülerinnen und Schüler können • Algorithmen und Datentypen entwerfen und in Programmen umsetzen; • Techniken zur Modularisierung einsetzen. • Grenzen des Rechnereinsatzes darlegen. **Leitidee 3: Problemlösen und Modellieren** Die Schülerinnen und Schüler • kennen Grundlegende Prinzipien beim Problemlösen und diese anwenden;	Strukturierte Datentypen Einfache Sortierverfahren Rekursion Einfache Suchverfahren Rekursion	Eindimensionale Felder, auch ein mehrdimensionales Feld als Beispiel z.B straight-insertion, bubblesort z.B. Morsecode einfache Rekursionen, z.B. Fakultät, Fibonacci, Türme von Hanoi Lineare Suche Analyse von Quelltexten
	• Können eine Lösung dokumentieren, präsentieren und vertreten.	praktische Grenzen der Berechenbarkeit Rechnen mit endlicher Stellenzahl kritisches Laufzeitverhalten	z.B. quadratischer, logarithmischer Aufwand

Fortgeschrittene objektorientierte Modellierung und Programmierung			
10	**Leitidee 3: Problemlösen und Modellieren** Die Schülerinnen und Schüler • kennen Konzepte der objektorientierten Modellierung; • Können Beziehungen zwischen Klassen und die Kommunikation zwischen Objekten analysieren und beschreiben; • Können ein Modell entwerfen und analysieren; • Kennen grundlegende Prinzipien beim Problemlösen und diese anwenden; • können ein Problem arbeitsteilig im Team lösen; • können den Problemlöseprozess strukturieren; • können eine Lösung dokumentieren, präsentieren und vertreten.	Klassendiagramme Vererbung Polymorphie Top-Down- und Bottom-Up-Vorgehensweise Modularisierung Problemanalyse, Modellbildung Implementierung und Bewertung der Lösung	Mit mehreren Klassen Aggregation und Assoziation „hat-Beziehung“, „kennt-Beziehung“ Projekt z.B. Geometrische Objekte und Abbildungen

Klasse 12			**Informatik 2-stündig**
	Kryptografie		
14	**Leitidee 6: Informatik und Gesellschaft** Die Schülerinnen und Schüler • kennen Aspekte der Datensicherheit; • können moderne Verschlüsselungsverfahren erläutern, beurteilen und anwenden; • entwickeln ein Bewusstsein für rechtliche und ethische Fragen der Nutzung von Information und Software; • Gewinnen Einsicht in die Verantwortung beim Entwurf und beim Einsatz informationsverarbeitender Systeme **Leitidee 2: Algorithmen und Daten** Die Schülerinnen und Schüler können • Grenzen des Rechnereinsatzes darlegen	Spuren im Netz, Angriffe aus dem Netz, Schutzmaßnahmen Verschlüsselungsverfahren Grenzen der Berechenbarkeit Digitale Signatur Schlüsselmanagement informationelle Selbstbestimmung Datenschutz	 z.B. Cäsar, Viginére, One-Time-Pad Crypto-Analyse RSA an einem Beispiel berechnet Zertifizierung

Automaten und formale Sprachen			
5	**Leitidee 3: Problemlösen und Modellieren** Die Schülerinnen und Schüler können • Abläufe mit Hilfe von Zustandsdiagrammen modellieren. **Leitidee 4: Sprachen und Automaten** Die Schülerinnen und Schüler • können zwischen Syntax und Semantik unterscheiden; • Kennen den syntaktischen Aufbau einer formalen Sprache und können einfache formale Sprachen in Syntaxdiagrammen und Grammatiken darstellen; • Können endliche Automaten zur Syntaxprüfung regulärer Sprachen einsetzen • können die Grenzen der endlichen Automaten und der algorithmischen Berechenbarkeit aufzeigen	Zustandsmodellierung Zustand, Übergang, Zustandsdiagramm Reguläre Sprachen Grammatiken Endlicher Automat endlicher erkennender Automat Syntaxdiagramme Grenzen endlicher Automaten theoretische Grenzen der Berechenbarkeit	Als Beschreibung von Sprachen, die nicht mit endlichen erkennenden Automaten formuliert werden können z.B. für eingeführte Programmiersprachen, Klammerterme Halteproblem bei der Turingmaschine z.B. Getränkeautomat

Rechneraufbau			
10	**Leitidee 6: Informatik und Gesellschaft** Die Schülerinnen und Schüler kennen • die geschichtliche Entwicklung der Rechenmaschinen und Informationstechnik im Überblick. **Leitidee 5: Wirkprinzipien von Informatiksystemen** Die Schülerinnen und Schüler • gewinnen Einsicht in den Aufbau und die Prinzipien der Arbeitsweise eines Rechners; • können das Zusammenwirken von Rechenwerk, Steuerwerk und Speicher erläutern.	Geschichtliche Entwicklung der Informatik Prinzip des von Neumann-Rechners Betriebssystem, Compiler, Maschinensprache	Rechenwerk, Steuerwerk, Speicher z.B. mit Hilfe von Microsim, COSI Aufbau eines Halbaddierers mit Hilfe von Hardware oder einer Simulationssoftware z.B. MMLogic Nur exemplarisch