

Bezug zu den Leitideen der Bildungsstandards und den Kompetenzen, welche die Schülerinnen und Schüler erwerben	Inhalte	Bemerkungen
Klasse 11		Informatik 4-stündig
Einführung in die objektorientierte Modellierung und Programmierung		
<p>Leitidee 3: Problemlösen und Modellieren Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • können zu realen Problemen ein passendes Modellierungs-verfahren auswählen und ihre Wahl begründen; • kennen Konzepte der objektorientierten Modellierung; • können Beziehungen zwischen Klassen und die Kommunikation zwischen Objekten analysieren und beschreiben; • können ein Modell entwerfen und analysieren. <p>Leitidee 2: Algorithmen und Daten Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> • Benutzerschnittstellen mit einfachen Komponenten gestalten; • Techniken zur Modularisierung einsetzen. <p>Leitidee 3: Problemlösen und Modellieren Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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. 	<p>OOM Top-Down-Entwurf Objekt, Klasse, Attribut, Methode einfache Datentypen, Variablenkonzept Klassendiagramme</p> <p>Geheimnisprinzip Zustand und Verhalten eines Objektes</p>	<p>Einsatz einer geeigneten Entwicklungsumgebung z.B. BlueJ oder JavaEditor Beginn mit einer einzelnen Klasse noch ohne Aggregation, Assoziation</p> <p>Hier werden die oben formulierten Methoden ausprogrammiert</p> <p>MVC-Modell Projektarbeit</p>

Kryptografie		
<p>Leitidee 6: Informatik und Gesellschaft Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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 <p>Leitidee 2: Algorithmen und Daten Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> • Grenzen des Rechnereinsatzes darlegen 	<p>Spuren im Netz, Angriffe aus dem Netz, Schutzmaßnahmen Verschlüsselungsverfahren, symmetrisch und assymetrisch</p> <p>Grenzen der Berechenbarkeit</p> <p>Digitale Signatur Schlüsselmanagement informationelle Selbstbestimmung Datenschutz</p>	<p>z.B. Cäsar, Viginère, One-Time-Pad Crypto-Analyse RSA an einem Beispiel berechnet</p> <p>Zertifizierung</p>

Abstrakte Datentypen		
<p>Leitidee 2: Algorithmen und Daten Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> • elementare und abstrakte Datentypen sowie Strukturen zur Ablaufsteuerung anwenden; • Algorithmen und Datentypen entwerfen und in Programmen umsetzen; • Techniken zur Modularisierung einsetzen 	<p>Strukturierte Datentypen</p> <p>Einfache Sortierverfahren</p> <p>Abstrakte Datentypen Liste, Schlange, Keller</p> <p>Binärbaum Rekursion als Lösungsprinzip</p> <p>Einfache Suchverfahren</p>	<p>Eindimensionale Felder als Übergang zu den ADTEn, auch ein mehrdimensionales Feld als Beispiel z.B. straight-insertion, bubblesort</p> <p>Implementierung mindestens eines ADTs in der eingeführten Programmiersprache z.B. Huffmann, Morsecode einfache Rekursionen, z.B. Fakultät, Fibonacci, Türme von Hanoi Lineare Suche</p>
<p>Leitidee 3: Problemlösen und Modellieren Die Schülerinnen und Schüler kennen</p> <ul style="list-style-type: none"> • grundlegende Prinzipien beim Problemlösen anwenden. 		

Fortgeschrittene objektorientierte Modellierung und Programmierung		
<p>Leitidee 3: Problemlösen und Modellieren Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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. 	<p>Klassendiagramme</p> <p>Vererbung Polymorphie</p> <p>Top-Down- und Bottom-Up-Vorgehensweise Modularisierung Problemanalyse, Modellbildung Implementierung und Bewertung der Lösung</p>	<p>Mit mehreren Klassen Aggregation und Assoziation „hat-Beziehung“, „kennt-Beziehung“</p> <p>Projekt z.B. Geometrische Objekte und Abbildungen</p>

Automaten und formale Sprachen		
<p>Leitidee 3: Problemlösen und Modellieren Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> • Abläufe mit Hilfe von Zustandsdiagrammen modellieren. <p>Leitidee 4: Sprachen und Automaten Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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 	<p>Zustandsmodellierung Zustand, Übergang, Zustandsdiagramm</p> <p>Reguläre Sprachen Grammatiken</p> <p>Endlicher Automat endlicher erkennender Automat Syntaxdiagramme Grenzen endlicher Automaten theoretische Grenzen der Berechenbarkeit</p>	<p>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</p>

Rechneraufbau		
<p>Leitidee 6: Informatik und Gesellschaft Die Schülerinnen und Schüler kennen</p> <ul style="list-style-type: none"> • die geschichtliche Entwicklung der Rechenmaschinen und Informationstechnik im Überblick. <p>Leitidee 5: Wirkprinzipien von Informatiksystemen Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • gewinnen Einsicht in den Aufbau und die Prinzipien der Arbeitsweise eines Rechners; • können das Zusammenwirken von Rechenwerk, Steuerwerk und Speicher erläutern. 	<p>Geschichtliche Entwicklung der Informatik Prinzip des von Neumann-Rechners</p> <p>Betriebssystem, Compiler, Maschinensprache</p>	<p>Rechenwerk, Steuerwerk, Speicher z.B. mit Hilfe von Microsim, COSI Aufbau eines Halbaddierers mit Hilfe von Hardware oder einer Simulationssoftware z.B. MMLogic</p> <p>Nur exemplarisch</p>