

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

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

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

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

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

Qualifikationsphase Q ₁ (ca. 70 60-Minuten-Stunden).	Hinweis. Die Themen in Q ₁ und Q ₂ können getauscht werden.
Q₁: Datenstrukturen in Java (10 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
Lineare Datenstrukturen (10 h) <ul style="list-style-type: none"> • Begriff Datenstruktur Anforderungen an eine Datenstruktur (DS) • Unterschied statische und dynamische DS • Statische DS Array <ul style="list-style-type: none"> ○ Eigenschaften und Standardoperationen für ein- und zweidimensionale Arrays ○ Modellierung und Implementierung von Anwendungen • Dynamische DS → Queue, Stack, List <ul style="list-style-type: none"> ○ Eigenschaften und Standardoperationen und Methoden der Datenstrukturen ○ Erläuterung von Problemstellungen, die nach dem FIFO- bzw. LIFO-Prinzip bearbeitet werden ○ Modellierung und Implementierung von Anwendungen unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List) 	<ul style="list-style-type: none"> • erläutern Operationen dynamischer linearer Datenstrukturen • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu • interpretieren Fehlermeldungen und korrigieren den Quellcode • stellen lineare Strukturen grafisch dar und erläutern ihren Aufbau • modellieren Klassen und dokumentieren Klassen • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie • untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts

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

Q₁: Kommunikation in Netzwerken (12 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
Menschen kommunizieren – ohne und mit Technik (1 - 2 h) <ul style="list-style-type: none"> • Wie läuft eine Kommunikation zwischen (2 oder mehr) Menschen ohne Technik ab? (Shannon-Weaver-Modell) • Anforderungen an technische Kommunikationen für eine fehlerfreie Übertragung • Geschichtliche Entwicklung 	<ul style="list-style-type: none"> • stellen Abläufe und Bedingungen von erfolgreichen Kommunikationsprozesse dar • übertragen diese Erkenntnisse auf die technische Kommunikation • lernen alte technische Kommunikationsformen kennen
Ohne Protokolle läuft nichts – Netzwerke (10 h) <ul style="list-style-type: none"> • Bestandteile eines Netzwerkes und deren Funktion • Verschiedene Topologien • Protokolle – Kommunikationsregeln • OSI-Schichtenmodell • TCP/IP Modell (Anwendungsschicht, Transportschicht, Internetschicht; optional: Netzzugangsschicht) • IPv4-Struktur (Netz- und Hostteil), Subnetting, optional: Klassen A, B, C • Simulation von Netzwerken mit Filius 	<ul style="list-style-type: none"> • erforschen das eigene (Schul-)Netz • skizzieren die Topologie an unserer Schule • lernen die Schichten des OSI-Modells (TCP/IP-Modells) kennen • kennen den Unterschied zwischen den Netzwerkklassen

Q₁: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten (33 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
<p>Nutzung von relationalen Datenbanken (16h)</p> <ul style="list-style-type: none"> • Aufbau von Datenbanksystemen und Grundbegriffe <ul style="list-style-type: none"> ○ Aufgaben und Eigenschaften eines Datenbanksystems ○ Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema ○ Problematisierung von Redundanzen, Anomalien und Inkonsistenzen • SQL-Abfragen <ul style="list-style-type: none"> ○ Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) ○ Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke) • Vertiefung an einem weiteren Datenbankbeispiel 	<ul style="list-style-type: none"> • erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung, • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage, • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren, • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen,

Q₁: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten (35 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
<p>Modellierung von relationalen Datenbanken (17 h)</p> <ul style="list-style-type: none"> • Datenbankentwurf durch ER-Diagramme <ul style="list-style-type: none"> ○ Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms ○ Erläuterung und Erweiterung einer Datenbankmodellierung • Entwicklung eines relationalen Modells aus einem Datenbankentwurf <ul style="list-style-type: none"> ○ Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln • Normalformen <ul style="list-style-type: none"> ○ Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) <p>Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</p>	<ul style="list-style-type: none"> • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten, • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar • modifizieren eine Datenbankmodellierung, • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema, • bestimmen Primär- und Sekundärschlüssel, • analysieren und erläutern eine Datenbankmodellierung, • erläutern die Eigenschaften normalisierter Datenbankschemata, • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften. • überführen Datenbankschemata in die 1. bis 3. Normalform

Q₁: Optional: Datensicherheit und Datenschutz (3 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
Datenschutz und Datensicherheit (3 h)	<ul style="list-style-type: none"> • nehmen das Recht auf informationelle Selbstbestimmung wahr und wenden die Gesetze zum Datenschutz an • 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

Qualifikationsphase Q ₂ (ca. 56 60-Minuten-Stunden)	Hinweis. Die Themen in Q ₁ und Q ₂ können getauscht werden.
Q₂: Kryptologie und Datensicherheit (20 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
Das Caesar-Verfahren (4 h) <ul style="list-style-type: none"> • Grundbegriffe der Kryptologie (Ver- und Entschlüsselung, Geheimtextalphabet, Klartextalphabet, ...) • Geschichte der Kryptologie • Modulare Addition und Subtraktion • Optional: Kryptoanalyse des Caesar-Verfahrens • Optional: weitere symmetrische Verschlüsselungsverfahren • Symmetrische und asymmetrische Verschlüsselung 	<ul style="list-style-type: none"> • kennen und wenden das Caesar-Verfahren als typische monoalphabetische Verschlüsselung an • brechen den Caesar-Code durch Ausschöpfen des Schlüsselraums • können symmetrische und asymmetrische Verschlüsselungsverfahren unterscheiden
Multiplikative Verschlüsselung (5 h) <ul style="list-style-type: none"> • Optional: Ver- und Entschlüsselung durch Modulare Multiplikation • Optional: Anwendung des Verschlüsselungsverfahrens auf große Zahlen (Blockchiffre) • Einführung des Prinzips der asymmetrischen Verschlüsselung (Problem der Schlüsselverteilung und der authentischen Kommunikation) • Erweiterter euklidischer Algorithmus zur Bestimmung der modularen Inversen 	<ul style="list-style-type: none"> • optional: führen die multiplikative Verschlüsselung (auch mit großen Zahlen) durchführen können • optional: werden mit dem Verfahren der Blockchiffrierung vertraut • wenden die Grundprinzipien der asymmetrischen Kryptographie an • optional: beurteilen die mangelnde Eignung der multiplikativen Verschlüsselung für die asymmetrische Kryptographie
Das RSA-Verfahren (6 h) <ul style="list-style-type: none"> • Chiffrierung durch modulares Potenzieren • Schlüsselgenerierung beim RSA-Verfahren • Einwegfunktionen • Nachweis der Korrektheit des RSA-Algorithmus 	<ul style="list-style-type: none"> • wenden die Ver- und Entschlüsselung nach dem RSA-Verfahren an • verstehen das Prinzip zur Bestimmung von (großen) Primzahlen • erfassen die Einweg-Eigenschaft der Multiplikation großer Primzahlen erfassen • generieren Schlüsselpaare • beurteilen die Sicherheit des RSA-Verfahrens und vollziehen die Korrektheit des RSA-Verfahrens

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

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

Q₂: Systementwicklung (15 h)	
Inhaltliche Schwerpunkte	Kompetenzen
	Die SuS
<p>Projektdurchführung</p> <ul style="list-style-type: none"> • abhängig vom Thema können auch Sachgebiete der anderen Jahrgangsstufen berührt werden • Themenwahl der Projektarbeit orientiert sich an den Interessen der Schülerinnen und Schüler • Auswahl: <ul style="list-style-type: none"> ○ DB-Entwurfsverfahren ○ Elemente des Software-Engineering (z.B. systematisches Testen, Programmdokumentation) 	<ul style="list-style-type: none"> • vertiefen und erweitern die in den Jahrgangsstufen E und Q1 erarbeiteten Grundlagen • wenden die grundsätzlichen Prinzipien der Projektarbeit an • planen von Projekten • arbeiten im Team Dokumentation der Projektarbeit