

Schulinterner Arbeitsplan Informatik Qualifikationsphase Schuljahr 2019/20

Auf den folgenden Seiten sind **tabellarisch** Module und Projekte dargestellt, die in der Qualifikationsphase zu unterrichten sind. Sie orientieren sich an der Arbeitsfassung des KC für die Sekundarstufe II, soweit dies bzgl. der Schwerpunkte des jeweiligen Jahrgangs passend erscheint. Die Nummerierung der prozess- und inhaltsbezogenen Kompetenzen ist der letzten Seite der oben genannten Arbeitsfassung entnommen.

Die Zuordnung der Module zu den Jahrgängen ist dabei in der folgenden Tabelle zusammengefasst:

	Jahrgang 12 (Abitur 2021)	Jahrgang 13 (Abitur 2021)
1. Halbjahr	QM3: Algorithmen und Datenstrukturen	QM 2: Endliche Automaten und formale Sprachen
2. Halbjahr	QM5: Informationen und Daten QPK1: Robotikprojekt QPK2: Programmieren im Team	QV: Vertiefung

QM1: Datenbanken sowie Datenschutz und Urheberrecht

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
2 Wochen	<ul style="list-style-type: none"> - beschreiben die Wirkungsweise grundlegender SQL-Abfragen zur Datenbankauswertung anhand eines konkreten Satzes von Relationen. (PK1.1, PK3) - formulieren einfache Abfragen und Verbundabfragen über mehrere Tabellen in SQL (IK2.4) - formulieren Abfragen an Datenbanken durch Verwendung von Aggregatfunktionen (IK2.4) 	Anhand der Datenbank Terra werden zunächst SQL-Abfragen geübt. Der Abschnitt ist entdeckend ohne theoretische Basis angelegt. Diese Phase kann mit dem anschließenden Theorieteil gemischt werden.	<ul style="list-style-type: none"> - Tabelle, Zeile, Spalte, Datensatz, Tupel - select from where - order by ASC/DESC - logische Operatoren (not, and, or, ...) - Vergleichsoperatoren (<, =, is not, like, ...) - Gruppenbildung: count/group by/having - Aggregatfunktionen: sum/avg/max/min... 	Datenbank Terra, LibreOffice Base
3-4 Wochen	<ul style="list-style-type: none"> - unterscheiden Datensatz, Attribut, Schlüsselattribut, Tabelle und Datenbank (PK1, PK3) - erläutern die Notwendigkeit, Datensätze eindeutig identifizieren zu können (Primärschlüssel) (PK1, PK3, IK1.5) 	In diesem Theorieteil werden entlang des vorangegangenen Praxisteils die relevanten Begriffe eingeführt. Der Umgang mit logischen Operatoren und Relationenalgebra wird geübt.	<ul style="list-style-type: none"> - Datenbankstruktur - Datenbankschema - Datenbankentwurf - Tabelle - Datensatz - Relationale Datenbank - Relation - Attribut/ Attributwert - Tupel - Primär-/ Fremdschlüssel - UND/ODER/NICHT... - Join - Project - Select 	Datenbank Terra
4-5 Wochen	<ul style="list-style-type: none"> - erläutern den grundlegenden Aufbau von ER-Diagrammen (PK1, PK3) - entwickeln ein ER-Modell für ein gegebenes System(PK1.4, IK1.3-5) 	Mit der Einführung des ER-Modells wird die Datenbank-Modellierung geübt. Anhand von Redundanz und Anomalien	<ul style="list-style-type: none"> - Entity, Relationship, Kardinalitäten - Normalformen nach Codd - redundante Infos 	Datenbank Terra, LibreOffice Base

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	<ul style="list-style-type: none"> - analysieren bzw. erweitern ein vorgegebenes ER-Modell bezüglich eines Anwendungsfall (PK1.3, PK4.1) - setzen ein ER-Modell in ein relationales Datenbankschema um. (IK1) - erläutern die Bedeutung von Kardinalitäten für die Beziehungen zwischen Entitäten (PK1, PK3, IK1.5) - nennen Beispiele für Einfüge-, Änderungs- und Löschanomalien (PK3, PK4.2) - untersuchen eine gegebene Datenbankmodellierung auf mögliche auftretende Anomalien (IK1.5) - analysieren eine Tabellenstruktur bezüglich der ersten, zweiten und dritten Normalform (PK1.3) - entwickeln ein relationales Datenbankschema unter Berücksichtigung der ersten, zweiten und dritten Normalform (PK1.4, IK1.3-5) 	<p>werden die verschiedenen Normalformen eingeführt und diskutiert.</p>	<ul style="list-style-type: none"> - atomar - voll funktional abhängig 	
2 Wochen	<p><u>Datenschutz und Urheberrecht</u></p> <ul style="list-style-type: none"> - erläutern die rechtlichen Rahmenbedingungen für den Umgang mit ihren persönlichen Daten, wie z. B. informationelle Selbstbestimmung und Datenschutzrichtlinien (PK3.1, PK3.3, PK3.4) - diskutieren die rechtlichen Aspekte bei der Speicherung, Auswertung und Veröffentlichung von Daten (IK4) - reagieren angemessen auf Risiken bei der Nutzung von Informatiksystemen (IK4.1) - nehmen Entscheidungsfreiheiten im Umgang mit Informatiksystemen wahr und handeln in Übereinstimmung mit gesellschaftlichen Normen (IK4.1) - benennen Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung (PK3) <p><u>Internet</u></p>	<p>Die Themen Datenschutz, Datensicherheit und Übertragungssicherheit werden an einem Vergleich von E-Mail und WhatsApp-Nachrichten behandelt. Persönlichkeitsrecht und Urheberrecht werden anhand einer Homepage-Veröffentlichung diskutiert. Darüber hinaus sind Suchmaschinen als Datensammler und Filesharing-Börsen als potentielle Raubkopieverteiler zu thematisieren.</p>	<p>Datenschutz, Datensicherheit, Übertragungssicherheit, Protokoll, Verschlüsselung, Metadaten, Urheber, Raubkopie. Viele Fachbegriffe, wie IP-Adresse, PK2P, Klartext, public key, etc. sind den Schülerinnen und Schülern aus anderen Modulen bekannt und werden hier nur verwendet.</p>	Internet, Iserv

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	<ul style="list-style-type: none"> - benennen verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise (IK3.1) - wenden Informatiksysteme zielgerichtet an und erschließen sich weitere Informatiksysteme (IK3) 			

QM2: Endliche Automaten und formale Sprachen

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
6 Wochen	<ul style="list-style-type: none"> - beschreiben den Aufbau und die Funktionsweise eines endlichen Automaten (DEA). (PK1.1) - beschreiben den Aufbau und die Funktionsweise eines endlichen Automaten mit Ausgabe (Mealy-Automat). (PK1.1) - entwickeln und implementieren Automatenmodelle in Form von Zustandsgraphen. (PK1.3-4, PK2.2, PK4.2-3) - analysieren die Funktionsfähigkeit eines durch einen Zustandsgraphen vorgegebenen Automaten. (PK2.2-3, PK3.1-3, IK3.1) - erläutern die Grenzen von endlichen Automaten bei der Analyse von Eingaben. (PK3.1-3, IK4.3) 	<p>Einstieg: Beschreibung realer Automaten:</p> <ul style="list-style-type: none"> - Videorekorder - Getränkeautomat - Ampelschaltung - Parkautomat <p>Zu JFLAP erhalten die Schülerinnen und Schüler ein Tutorium mit vielen Beispielen und Aufgaben zur Bearbeitung. Sie sollen sich die Software nach dem Konzept „Learning by doing“ selbst aneignen. Im Anschluss wird die Theorie aufgearbeitet.</p>	<ul style="list-style-type: none"> - DEA/Mealy-Automat - Akzeptor/Transduktor - NEA - Alphabet (Eingabe- und Ausgabealphabet) - Zustand/Zustandsmenge - Start- und Endzustand - Übergangsfunktion - Ausgabefunktion - Zustandsdiagramm (Zustandsgraph) - Akzeptierender Zustand - Akzeptierte Sprache 	<p>JFLAP : Dient zum Erzeugen und Testen von Automaten.</p>
6 Wochen	<ul style="list-style-type: none"> - nennen Eigenschaften formaler Sprachen an verschiedenen Beispielen. (PK3.1, PK3.3) - analysieren und beschreiben die von einer Grammatik erzeugte Sprache. (PK4.2, IK3.1, IK3.3) - erstellen bzw. entwickeln Grammatiken für formale 	<p>Ziel dieses Abschnittes ist es, Zusammenhänge zwischen endlichen Automaten und regulären Sprachen bzw. regulären Grammatiken durch</p>	<ul style="list-style-type: none"> - Wort/Sprache - Syntax/Semantik/ Pragmatik - Reguläre Sprachen - Grammatik 	<p>JFLAP Railroad Diagram Generator: erzeugt aus einer Grammatik in der EBNF ein entsprechendes Syntaxdiagramm.</p>

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	Sprachen bzw. für ein gegebenes Problem. (PK4.3, IK3.1-3) - erläutern den Zusammenhang zwischen regulären Grammatiken und endlichen Automaten. (IK4.3) - analysieren und beschreiben die Syntax einer Sprache mit Hilfe eines Syntaxdiagramms (PK4.2, IK3.1, IK3.3) - stellen die Grammatik einer Sprache in der EBNF dar. (PK1.4)	Experimente (JFLAP) herauszufinden.	<ul style="list-style-type: none"> - Nichtterminalsymbole - Terminalsymbole - Startsymbol - Produktionen (Regeln) - Wort ableiten - Syntaxdiagramm - Sequenz - Alternative - Option - Iteration - Rekursion 	

QM3: Algorithmen und Datenstrukturen

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
4 Wochen	<u>Klassen und Objekte</u> - verwenden gegebene Klassen/Objekte zur Realisierung eigener Programme. (PK2.2, PK2.4, PK4, IK1.3-4) - entwerfen und implementieren Klassen/Objekte zur Realisierung eigener Programme. (PK2.2, PK2.4, PK4, IK1.3-4) - stellen Klassen/Objekte und deren Beziehungen durch UML- Klassen-/Objektdiagramme dar. (PK1.3-4, PK3.2)	<ul style="list-style-type: none"> - Strukturelle Aspekte bei der Konstruktion von Programmsystemen stehen im Vordergrund. - Analyse und Modellierung folgen dem objektorientierten Ansatz. - Projektorientierte Vorgehensweise als Vorbereitung auf ein Programmierprojekt. 	<ul style="list-style-type: none"> - Klasse/Objekt/Instanz einer Klasse - Attribut(-wert)/Datentyp - Methoden - Methodenaufruf/ Punktnotation - Klassendiagramm - Beziehungen und Kardinalitäten/ Ist-, Hat- und Kennt-Beziehungen zwischen Klassen - Binäre Assoziation - Aggregation/Komposition - Generalisierung/Vererbung - Polymorphismus - Sichtbarkeit (private, protected und public) 	<ul style="list-style-type: none"> - JavaEditor - UMLet: UML-Editor
3 Wochen	<u>Abstrakte Datenstrukturen (Lineare Strukturen)</u>	Es werden mindestens die ADTs	<ul style="list-style-type: none"> - Ein- und zweidimensionale 	Javaeditor

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	<ul style="list-style-type: none"> - erläutern das Prinzip, mehrere Objekte des gleichen Typs in Reihungen zu verwalten, zu suchen und zu sortieren. (IK1.3, PK2, PK3, PK4) - verwenden ein- und zweidimensionale Arrays zur Implementierung ihrer Programmsysteme. (IK1.3, PK2, PK4) - erläutern das Prinzip der Datenstrukturen Stapel, Schlange und dynamische Reihung (PK3) - realisieren bzw. implementieren alle Funktionalitäten der Datenstrukturen Stapel/Schlange mit Hilfe einer Objektorientierten Sprache (Java). (PK4, IK1.3, IK1.4, IK2.2) - verwenden die Datenstrukturen Stapel, Schlange und dynamische Reihung zur Realisierung eigener Programme (PK4, IK1.3) 	<p>Stapel und Schlange auf Basis einer selbst implementierten linearen Liste in Form einer Klasse erstellt.</p> <p>Als Anwendung für die dynamische Reihung wird BubbleSort und zur Differenzierung ein weiteres Sortierverfahren implementiert (Quicksort oder Mergesort).</p> <p>Zur weiteren Differenzierung lassen sich Rechenterme in der Präfixnotation mit Hilfe eines Stapels darstellen.</p>	<p>Arrays</p> <ul style="list-style-type: none"> - BubbleSort - Stapel (Stack) - Stapel(), entnehmen(), ablegen(), inhaltGeben(), istLeer() - Last-In-First-Out-Prinzip - Schlange (Queue) - First-In-First-Out-Prinzip - Schlange(), entnehmen(), anhängen(), inhaltGeben(), istLeer() 	<p>Nessi (Nassi-Shneiderman-Diagramm)</p>
6 Wochen	<p><u>Grundlagen der Algorithmik</u></p> <ul style="list-style-type: none"> - benennen die Grundstrukturen eines Algorithmus. (PK1.1-2) - entwerfen einen Algorithmus unter zielgerichteter Verwendung der elementaren Kontrollstrukturen. (PK2.1) - erläutern das Prinzip der Speicherung von Werten in Variablen und verwalten Daten anhand von Variablen. (IK1.2, PK3.2, IK2.1) - protokollieren die Veränderungen von Variablenzuständen, wenn mehrere Zuweisungen (bzw. sonstige Anweisungen) ausgeführt werden. (IK1.2, PK3.2, IK2.1) - unterscheiden zwischen lokalen und globalen Variablen und zwischen primitiven und Referenz-Datentypen. (PK2.4, PK3, PK4.3) - verwenden Operationen, unter Beachtung der Übergabeparameter und Rückgabewerte, zur strukturierten Implementierung von Algorithmen. (PK2.4, PK3) - erläutern das Konzept der Rekursion als Aufruf der 	<p>Größere Teile sind Wiederholungen des Programmiermoduls der Einführungsphase.</p> <p>Hinzu kommen Variablentypen, Datentypen, Methodenaufrufe und Rekursion.</p> <p>Rekursion wird dabei zumindest an den Beispielen Fakultät und GGT untersucht. Sie tritt im Rahmen der Differenzierung auch beim Sortieren auf.</p> <p>Zum Nachvollziehen der Rekursionen werden Trace-Tabellen eingesetzt.</p>	<ul style="list-style-type: none"> - Anweisung, Sequenz, Schleifen und Verzweigung - Programmablaufplan/ Struktogramm/ Pseudocode - Deklaration und Initialisierung von Variable, Zuweisungen, Typumwandlungen(Explizit/implizit), Trace-Tabelle - byte, short, int, long, float, double, char und boolean. - Bezeichner, Sichtbarkeit und Lebensdauer, überladen von Methoden. 	<p>Javaeditor</p> <p>Nessi (Nassi-Shneiderman-Diagramm)</p>

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	<p>eigenen Methode mit anderem Parameter und erkennen es als eine Alternative zur Verwendung von Schleifen. (PK2, PK4)</p> <ul style="list-style-type: none"> - implementieren rekursive Algorithmen in gegebenen Zusammenhängen wie z.B. (Fakultät, Fibonacci-Zahlen, Berechnung des GGT zweier Zahlen). - erläutern die Strategie „Teile und herrsche“ beim Entwurf rekursiver Algorithmen. (PK2, PK4, IK2.1) - vervollständigen, korrigieren und verbessern einen gegebenen Algorithmus. (PK1-4) - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen. PK1-4) 			

QM4: Schaltnetze

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
6 Wochen	<ul style="list-style-type: none"> • analysieren und erläutern den Zusammenhang von Information und Daten (duales Zahlensystem, boolesche Algebra und grundlegende mathematische Operationen) (PK1.4, IK1.1) • stellen Informationen in unterschiedlicher Form dar (Codierung von Zahlen und Zeichen) und interpretieren Daten im Kontext der repräsentativen Information. (IK1.2, IK1.4) • erarbeiten die logischen Verknüpfungen AND, OR, NOT und XOR mit ihren Wahrheitstabellen (IK1.4) • verdeutlichen die Verarbeitung binär dargestellter Daten in einfachen Fällen mit logischen Schaltungen und/oder mit einer geeigneten Simulations-Software (logisim) (IK1.3, PK3.2, PK3.4, PK5.1) • verstehen die Zusammensetzung von Schaltnetzen aus Schaltern mit Eingangs- und Ausgangsleitungen (PK1.4, IK1.4) • verwenden grundlegende Operationen zum Zugriff auf die Bestandteile strukturierter Daten (PK3.2) • verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (PK3.1) • beschreiben Schaltnetze durch Eingangs- und Ausgangsbelegungen in Form einer Schaltwerttabelle (PK1.3, PK3.2). • können den Zusammenhang zwischen Ein- und Ausgängen funktional durch einen Term beschreiben und vereinfachen (Schaltalgebra) (PK1.2, PK1.5). • stellen Minterme als logische UND- 	<p>Als Einführungsbeispiel dient eine Schranke, die durch zwei Lichtschranken gesteuert wird. Nach einer Softwarelösung wird eine analoge Schaltung entworfen. Anschließend wird die Boolesche Algebra theoretisch bearbeitet. Terme werden zunächst durch Ausprobieren, danach systematisch mit KV-Diagrammen optimiert. Aus Halb- und Volladdierern wird ein 4- und/oder ein 8-Bit-Addierer zusammengesetzt und mit logisim simuliert. Simulationen sollten auch an anderer Stelle durchgeführt werden. Dabei wird die Zweierkomplementdarstellung bzgl. der Überträge berücksichtigt. Zur Binnendifferenzierung können als ergänzende Themen ALU und rückgekoppelte Schaltnetze herangezogen werden.</p>	<ul style="list-style-type: none"> • Binäres (Duales) System, Boolesche Algebra • Schaltnetz, Schalter, Eingangs- und Ausgangsleitungen • Eingangs- und Ausgangsbelegung • Schaltvariablen, Minterm • Logikgatter bzw. Gatter, Gatternetz • Spannungszustand • Logische Operatoren (UND, ODER und NICHT) • Wahrheits-, Funktions-, Schaltwert- oder auch Zustandstabelle • disjunktive Normalformen • Gesetze der Schaltalgebra (Kommutativ-, Assoziativgesetz, ...) • Termvereinfachung • Das Verfahren von Quine-McCluskey • Halbaddierer, 	logisim, Java oder Arduino IDE

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	<p>Verknüpfung einer Kombination von Schaltvariablen dar (PK3.1, PK3.2).</p> <ul style="list-style-type: none"> • verstehen einen Term als ODER-Verknüpfung lauter Minterme und geben die disjunktiven Normalformen einer Schaltfunktion anhand einer Schaltwerttabelle an (PK3.1, PK3.2). • entwickeln einfache Schaltungen durch systematische Vereinfachung von Schaltfunktionen (PK1.3, PK1.4, IK1.4). • können Gesetze der Schaltalgebra durch Aufstellung von Schaltwerttabellen verifizieren (PK1.3, PK1.4, IK1.4). • können Terme durch Anwendung der Gesetze der Schaltalgebra oder systematisch nach dem Verfahren von Quine-McCluskey oder anhand des Karnaugh-Veitch-Diagramms vereinfachen (IK2.1, PK1.3, PK1.4). • können einfache Schaltnetze (Halbaddierer, Volladdierer, Multiplexer,...) entwickeln und aus diesen ein komplexeres Schaltnetz wie Paralleladdierer zusammensetzen (IK3.2, IK3.4). • können vorgegebene Gatterdarstellungen bzw Schaltungen verstehen, analysieren und ihre zugehörigen Schaltfunktionen zuordnen (PK1.2, PK1.4, PK1.5, IK3.1). • überprüfen die Korrektheit einer vereinfachten Schaltfunktion durch ein Programm (PK2.1, PK2.2, PK2.3, PK2.4). • setzen eine vorgegebene Schaltfunktion in eine Gatterdarstellung um und simulieren diese mit Hilfe der Software logisim und überprüfen ihre Korrektheit (PK3.1, PK3.2, PK3.3, PK3.4). 		<p>Volladdierer, 4-Bit-Addierer, Multiplexer, Demultiplexer, ALU, Paralleladdierer,...</p> <ul style="list-style-type: none"> • boolesche Algebra Aussagenlogik • KV-Diagramm • Zweierkomplement (Dual-Darstellung negativer Zahlen) 	

QM5: Informationen und Daten

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
12 Wochen	<p><u>Kryptologie</u></p> <ul style="list-style-type: none"> • beschreiben die Bausteine einer Geheimschrift anhand der Betrachtung und Anwendung bekannter Geheimschriften, u.a. die Geheimschrift von Polybius (PK1.2, PK1.3, PK1.4). • beschreiben das Prinzip der Transposition und der Substitution zur Verschlüsselung von Daten (PK1.2). • Implementieren monoalphabetische Verfahren, u.a. Skytale und Caesar-Verfahren (PK2.1, PK2.2, PK2.3, IK2.2). • beschreiben das Prinzip der polyalphabetischen Substitution, u. a. am Beispiel des Vigenère-Verfahrens (PK1.2). • Entwerfen und implementieren polyalphabetische Verfahren, u.a. Vigenère Verfahren (PK2.1, PK2.2, PK2.3, IK2.2). • beurteilen die Sicherheit eines gegebenen symmetrischen Verschlüsselungsverfahrens. <p><u>Codierung und Übertragung von Daten</u></p> <ul style="list-style-type: none"> • beschreiben grundlegende Codierungen von Daten, u. a. Dualzahlen, ASCII, RGB-Modell (PK3.3, IK1.1,)). 	<p>Als Einstieg in das Thema Kryptologie wird den SUS ein erster Einblick in historische und moderne kryptologische Verfahren durch einen Brickfilm vermittelt.</p> <p>Zum Brickfilm : stemue.de/</p> <p>Zum Ver- bzw. Entschlüsseln von Klar- bzw. Geheimentexten sowie für die Häufigkeitsanalyse werden von SuS selbst entwickelte Byob Programme oder Cryptool 1.4.42 benutzt.</p> <p>Für eine schnellere Häufigkeitsanalyse ist CrypTool besser geeignet.</p>	<ul style="list-style-type: none"> • Geheimschrift, Codierung, Klartext, Chiffrieren, Geheimtext, Dechiffrieren. • Geheimschrift von Polybius • Kryptosystem, Schlüssel, Ver- und Entschlüsselung, Kryptographie. • Skytale von Sparta • Transposition und Substitution • Caesar-Verfahren • Mono- und polyalphabetische Kryptosysteme, Vigenère-Verfahren, One-Time-Pad • Kryptoanalyse, Kryptologie, Häufigkeitsanalyse, Häufigkeitsverteilung, Bi- und Trigramme, Kasiski- und Friedman Test • symmetrisches bzw. 	<ul style="list-style-type: none"> • Scratch/Byob • Java/Python • Caesar-Scheibe • Arduino zum Übertragen von Daten • CRYPTOOL 1.4.41, erhältlich hier: https://www.cryptool.org/de/ct1-downloads • Das kostenlose Buch CT-Book-de über die in Cryptool verwendeten Kryptographischen Verfahren: erhältlich hier: file:///C:/Users/dogin/Desktop/Alle%20Dateien/Einführungsphase/Informatik/01-%20Grundlagen%20der%20Algorithmik%20Statische%20und%20dynamische%20Datenstrukturen/CT-Book-de.pdf

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
	<ul style="list-style-type: none"> • beschreiben Möglichkeiten, Daten zu komprimieren, u. a. Lauflängencodierung, Huffman-Codierung (PK1.1, IK1.1). • entwerfen und implementieren ein Kompressionsverfahren zu einem gegebenen Sachverhalt (PK2.1, PK2.2, PK2.3, IK1.1, IK2.3). • entwerfen und implementieren ein Protokoll zur Übertragung von Daten über einen Kommunikationskanal (IK1.1, IK2.3, IK3.4). • erläutern die Vor- und Nachteile verlustfreier und verlustbehafteter Kompression von Daten (PK2.3). • erläutern Möglichkeiten der Fehlererkennung und der Fehlerkorrektur bei der Datenübertragung, u. a. Paritätsbit, (7,4)-Hamming-Code (PK2.3, IK1.1, IK2.3). <p><u>Datenschutz und Datensicherheit</u></p> <ul style="list-style-type: none"> • Vergleiche QM1: Datenbanken sowie Datenschutz und Urheberrecht. 		asymmetrisches Verfahren. <ul style="list-style-type: none"> • Alphabet, Wort, binäre Darstellung, ASCII-Code. • Digitalisierung, Komprimierung mit und ohne Verlust, Dekomprimierung, Codewort, Präfixfreie Kodierung, Baumdiagramm, Huffman-Codierung. • EAN-Code, Prüfziffer bzw. Prüfsymbol oder Prüfbit. • K-fehlererkennend, k-fehlerkorrigierend, Abstand einer Komprimierung, • Paritätsbit, Hamming-Code 	

QPK1: Robotikprojekt

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
5 Wochen	<ul style="list-style-type: none"> - planen, organisieren und leiten selbstständig Projektarbeit hinsichtlich einer Teamarbeit. (PK3) - verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht. (PK3.1) - verwenden selbstständig Fachtexte, Dokumentationen und Hilfesysteme. (PK3.2) - setzen netzbasierte Kommunikations- und Kooperationssysteme in der Gruppenarbeit ein. (PK3) - dokumentieren, visualisieren, präsentieren und verteidigen Ergebnisse der Teamarbeit (PK3) - diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen (IK3.1) - entwickeln, implementieren, testen, reflektieren, beurteilen und validieren ihre eigenes Modell bzw. das einer anderen Gruppe (PK1.4, PK2.4, IK3.2, IK3.3). - entwerfen Algorithmen und stellen diese geeignet dar. (PK2.1) - analysieren, erläutern und vergleichen Algorithmen und beurteilen sie hinsichtlich ihrer Effizienz (PK2.3) - wenden die Phasen des Problemlösungsprozesses (informelle Problembeschreibung, formale Modellierung, Implementierung und Realisierung, Bewertung und Modellkritik) an. (PK4, IK3) - zeigen Grenzen des Problemlösens mit Informatiksystemen auf und beachten diese. (IK4) 	<p>Die Schülerinnen und Schüler bauen einen Roboter zusammen und programmieren ihn so, dass er ein unbekanntes Labyrinth ohne Berührung der Wände vom vorgegebenen Startpunkt bis zum Ausgang möglichst schnell durchfährt und danach zum Startpunkt zurück fährt.</p> <p>Der Aufbau wird mit der vorhandenen Mikrocontroller-Hardware (derzeit Arduino, Motorshield, etc.) realisiert.</p>	<ul style="list-style-type: none"> - künstliche Intelligenz - Autonomer Roboter - Mikrocontroller - Sensoren: <ul style="list-style-type: none"> - Infrarotsensor - Ultraschallsensor - Tastender Sensor - Abstandssensor - Aktoren: <ul style="list-style-type: none"> - Gleichstrommotor - Servomotor 	<ul style="list-style-type: none"> - Mikrocontroller Arduino UNO - Arduino Motor Shield - Chassis - Sensoren, Aktoren, Zubehör - Arduino IDE

QPK2: Programmieren im Team

Zeitraum	Kompetenzen Die Schülerinnen und Schüler ...	Hinweise/Bemerkungen	Fachsprache	Medien & Methoden fächerübergreifende Bezüge
5 Wochen	<ul style="list-style-type: none"> - planen, organisieren und leiten selbstständig Projektarbeit hinsichtlich einer Teamarbeit. (PK3) - verfügen über eine angemessene Fachsprache und verwenden sie sachgerecht. (PK3.1) - verwenden selbstständig Fachtexte, Dokumentationen und Hilfesysteme. (PK3.2) - setzen netzbasierte Kommunikations- und Kooperationssysteme in der Gruppenarbeit ein. (PK3) - dokumentieren, visualisieren, präsentieren und verteidigen Ergebnisse der Teamarbeit (PK3) - diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen (IK3.1) - entwickeln, implementieren, testen, reflektieren, beurteilen und validieren ihre eigenes Modell bzw. das einer anderen Gruppe (PK1.4, PK2.4, IK3.2, IK3.3). - entwerfen Algorithmen und stellen diese geeignet dar. (PK2.1) - analysieren, erläutern und vergleichen Algorithmen und beurteilen sie hinsichtlich ihrer Effizienz (PK2.3) - wenden die Phasen des Problemlösungsprozesses (informelle Problembeschreibung, formale Modellierung, Implementierung und Realisierung, Bewertung und Modellkritik) an. (PK4, IK3) - zeigen Grenzen des Problemlösens mit Informatiksystemen auf und beachten diese. (IK4) 	<p>Die Schülerinnen und Schüler sollen in Gruppen je ein Programmierprojekt in einer objektorientierten Sprache (i. d. R Java) umsetzen. Der Fokus liegt auf der Projektorganisation und -dokumentation. Innerhalb einer Gruppe sollen GUI-Entwicklung und Definition der Datenstrukturen von unterschiedlichen Teilgruppen ausgeführt werden. Es ist mindestens ein Koordinator pro Gruppe festzulegen. Das Projekt kann in einem Spiel oder einem Datenverwaltungsprogramm bestehen. Alle Gruppen sind dazu verpflichtet, in regelmäßigen Abständen über den Entwicklungsstand zu berichten (Entwurf+Dokumentation). Die Schülerinnen und Schüler werden ermuntert, das Projekt zur Wiederholung von eingeführten Datenstrukturen zu nutzen. Sie sollen nach Möglichkeit nicht in ihrer Wahl eingeschränkt aber bzgl. der Realisierbarkeit beraten werden.</p>	<ul style="list-style-type: none"> - SQL - Datenbank - OOP (Klassen/Objekte) - GUI - Datenstrukturen (Array/Liste/ Stapel/Schlange/Binärbaum) 	<ul style="list-style-type: none"> - Eclipse/BlueJ/JavaEditor - LibreOffice Base oder MySql oder SQLite - StarUML - Zur Dokumentation und Projektplanung geeignete Software (Die Wahl aus einigen Alternativen wird den Schülern überlassen).

QV: Vertiefung

Die Schülerinnen und Schüler erstellen mit Unterstützung der Lehrkraft eine Liste der Themen, die wiederholt oder vertieft behandelt werden sollen. Ausgehend von dieser Liste werden die behandelten Themen mit Hilfe von Abitur-relevanten Aufgaben bearbeitet.