

Inhaltsverzeichnis

Über diesen Arbeitsplan.....	2
Themenübersicht.....	3
Kurzdarstellung der einzelnen Themen.....	4
Jahrgang 9.....	4
Jahrgang 10.....	4
Einführungsphase.....	5
Qualifikationsphase.....	6
Bezug zum Medienbildungskonzept.....	9
Konkrete Medienbildungsinhalte der einzelnen Jahrgänge.....	13
Bezug zum Berufsorientierungskonzept.....	15
Sprachsensibler Unterricht.....	17
Beispiel für eine überarbeitete Fassung eines Unterrichtsinhalts.....	18
1. Ursprüngliche Variante.....	18
2. Erste Überarbeitung.....	20
3. Zweite Überarbeitung.....	23
Inhaltliche Umsetzung der curricularen Vorgaben nach Jahrgängen.....	27
Inhalte Jahrgang 9.....	27
Inhalte Jahrgang 10.....	28
Inhalte der Einführungsphase.....	30
Inhalte Jahrgang 12.....	32
Inhalte Jahrgang 13.....	35
Leistungsbewertung (Beschluss der Fachkonferenz vom 24.04.2022).....	36

Über diesen Arbeitsplan

Informatik kann in der Sek. 2 am GSG ab der Einführungsphase gewählt werden und findet in der Sek. 1 epochal in den Jahrgangsstufen 9 und 10 statt. Es ist in der E-Phase den Naturwissenschaften Biologie, Physik und Chemie gleichgestellt, d.h. die Schülerinnen und Schüler müssen drei aus diesen vier Fächern jeweils mit zwei Wochenstunden in der E-Phase belegen. Informatik kann in der Oberstufe nur als Prüfungsfach gewählt werden, wenn in der E-Phase Informatik belegt wurde.

Das Fach Informatik ist, wie andere Fächer auch, dem Bildungsauftrag verpflichtet. Dies wird im, ab dem Abiturjahrgang 2021 gültigen Kerncurriculum für die Sekundarstufe II genauer ausgeführt.

„In einem dem Bildungsauftrag verpflichteten Unterricht, wie er im Niedersächsischen Schulgesetz verankert ist, müssen zwei wesentliche Aspekte Berücksichtigung finden. Zum einen sollen Schülerinnen und Schüler befähigt werden, sich in einer dynamischen Gesellschaft unabhängig von ihrem Geschlecht oder sozioökonomischem Hintergrund selbstbestimmt und freiheitlich zu entwickeln. Zum anderen soll ihnen bewusst sein, dass sie durch ihr Handeln die Gesellschaft mitgestalten und dass sie für deren zukünftige Ausprägung Verantwortung übernehmen.

Die hohe Dynamik der Informationstechnologie sowie ihr stetig wachsender Einfluss auf gesellschaftliche und kulturelle Entwicklungen und die Lebenswirklichkeit erfordern einen selbstständigen, kompetenten und selbstbewussten Umgang mit ihr. Nur dadurch wird den Schülerinnen und Schülern ermöglicht, die gegenwärtigen und zukünftigen Auswirkungen dieser Technologien einzuschätzen. Sie werden in die Lage versetzt, aktiv, kreativ und gestaltend an der gesellschaftlichen Entwicklung teilzunehmen. Dies gilt insbesondere vor dem Hintergrund, dass in einer Informationsgesellschaft Erfolg in zunehmendem Maße nicht nur davon abhängt, was oder wie viel man weiß, sondern auch davon, ob man in der Lage ist, kreativ zu denken und zu handeln.

In allen Berufen und Wissenschaften haben sich die Arbeitsweisen durch die Methoden der Informatik und den Einsatz informatischer Systeme grundlegend verändert. Dies erfordert ein vertieftes Verständnis informatischer Zusammenhänge, um über das reine Anwenden hinaus, kreativ, reflektiert und zielgerichtet mit Informatiksystemen arbeiten zu können. Die dazu notwendigen Verfahren und Kenntnisse werden insbesondere durch den Informatikunterricht vermittelt.

Grundlegende informatische Methoden wie das Strukturieren, die systematische Analyse komplexer Systeme und das Modellieren fördern und fordern die Abstraktionsfähigkeit, sowie kreatives und strukturiertes Denken. Die Implementierung von Modellen und Lösungsideen als Programm oder technische Konstruktion macht sie überprüfbar, erlebbar und greifbar. Die konkrete Umsetzung als Produkt ermöglicht es, Modelle und Verfahren auf der Basis vorgegebener Funktionalitäten zu beurteilen. Dies fördert die Fähigkeit, sich konstruktiv und kritisch mit eigenen und fremden Ideen auseinanderzusetzen.

Die Bearbeitung komplexerer Problemstellungen im Rahmen von Teamarbeit erfordert es, klare Schnittstellen zu definieren und sich an Vereinbarungen zu halten, um zielorientiert zu arbeiten. In projektorientierten Arbeitsphasen durchlaufen die Schülerinnen und Schüler einen schöpferischen, kreativen Prozess, an dessen Ende ein Produkt steht. Sie erleben sich dabei als selbstwirksam und konstruktiv.

Medienkompetenz, wie sie der Informatikunterricht in besonderer Weise vermittelt, ist Voraussetzung für die Teilhabe an der Informations- und Wissensgesellschaft sowie an demokratischen Prozessen der Meinungsbildung. Erst ein grundlegendes technisches Verständnis von Informationssystemen befähigt dazu, Chancen und Risiken für die Gesellschaft abzusätzen und entsprechend verantwortungsvoll zu handeln.

Das Fach Informatik thematisiert soziale, ökonomische, ökologische und politische Phänomene und Probleme der nachhaltigen Entwicklung und trägt dazu bei, wechselseitige Abhängigkeiten zu erkennen und Wertmaßstäbe für eigenes Handeln sowie ein Verständnis für gesellschaftliche Entscheidungen zu entwickeln. Mit dem Erwerb spezifischer Kompetenzen wird im Unterricht des Faches Informatik der Bezug zu verschiedenen Berufsfeldern hergestellt. Die Schule ermöglicht es damit den Schülerinnen und Schülern, Vorstellungen über Berufe und über eigene Berufswünsche zu entwickeln, die über eine schulische Ausbildung, eine betriebliche Ausbildung, eine Ausbildung im dualen System oder über ein Studium zu erreichen sind. Der Fachunterricht leistet somit auch einen Beitrag zur Berufsorientierung, ggf. zur Entscheidung für einen Beruf. “ (*Niedersächsisches Kultusministerium, KC Informatik Sek I, Jahrgänge 5-10*)

Der Unterricht findet in einem der drei Fachräume If1, If2 oder If3 statt (Informatikräume). Somit stehen im Unterricht die entsprechenden Software- und Hardwarewerkzeuge, sowie Entwicklungsumgebungen zur Verfügung. Dem projektorientierten Arbeiten, als für den Informatikunterricht typischen Arbeitsform, wird in allen Jahrgängen Raum gegeben. Die Ergebnisse fließen jeweils in die Leistungsbewertung des entsprechenden Halbjahres ein.

Daniel Hoherz, OStR (Fachobmann Informatik), Mai 2024

Themenübersicht

Jahrgang 9

1. Algorithmisches Problemlösen und Automatisierte Prozesse - Algorithmisieren und Implementieren, auch durch Nutzung des Calliope
2. Computerkompetenz - Aufbau von Computersystemen
3. Computerkompetenz - Speichern von Daten
4. Informationen und Daten (Codierungstheorie)

Jahrgang 10

1. Daten und ihre Spuren - Aufbau von Netzwerken mit Schwerpunkt Internet
2. Algorithmisches Problemlösen - Algorithmisieren und Implementieren, auch durch Nutzung des Calliope
inklusive: Informatik und Gesellschaft - Auswirkungen von Informatiksystemen auf die Gesellschaft und Interessenabwägung bei der Ausgestaltung von Informatiksystemen
3. Daten und ihre Spuren - Verwaltung von Daten

Einführungsphase

1. Informationen und Daten (Codierungstheorie)
2. Algorithmen und Datenstrukturen mit Processing (Java)
3. Automatisierte Datenanalyse
4. Datenschutz
5. Kryptologie

Qualifikationsphase

Q1

1. Algorithmen und Datenstrukturen mit Java
2. Kryptologie und Codierungstheorie
3. Datenbanken

Q2

1. Theoretische Informatik, Datenanalyse und KI
2. Vertiefung Algorithmen und Datenstrukturen
3. Vertiefungen (variabel)

Kurzdarstellung der einzelnen Themen

Jahrgang 9

Die Schülerinnen und Schüler werden mit Hilfe eines **externen Mikrocontrollers**, dem Calliope, in das **algorithmische Problemlösen** mit einer grafischen Programmiersprache, OpenRoberta eingeführt. Nach einer Einstiegsphase zum Kennenlernen der Programmierumgebung und der Funktionalität des Calliopes im Kontext "Smartes Gewächshaus", werden kleinere Projekte zum Gewächshaus für verschiedene Aspekte algorithmisch umgesetzt. Anschließend entwickeln die Lernenden optional ein eigenes Projekt, welches mit Hilfe des Calliope modelliert werden soll.

Haben die Lernenden die Grundlagen von Algorithmen und Computersystemen (Kontrollstrukturen, EVA-Prinzip) kennengelernt, schauen sie sich noch eine weitere grafische Programmiersprache, Scratch, an. Die algorithmischen Kenntnisse werden hier mit Variablen in der Programmiersprache Scratch vertieft, welches den Folgeunterricht der Klasse 10 vorbereitet.

Das zweite Thema des Jahrgangs 9 ist eine **Einführung in die Codierungstheorie**. Nach einem Überblick zu den wichtigsten Komponenten eines gängigen Computersystems und Speichermedien / -orte entdecken die Lernenden notwendige Absprachen (u. a. einheitliche Codierung, eindeutige Decodierbarkeit, Fehlererkennung,...), die bei jeglicher Kommunikation getroffen werden müssen. Es folgt ein Überblick über das Dualsystem und der ASCII-Codierung, sowie ein Vergleich von zwei Codierungen (n-bit-Repetitionscode und ein verlustbehaftetes Bildkompressionsverfahren), um Anwendungsbereiche der Codierungstheorie zu entdecken. Abschließend werden noch eine weitere im Alltag genutzte Codierung (RGB, Farbbildcodierung, Braille, EAN) bearbeitet und mit dem Calliope ein Kommunikationsprotokoll für die ASCII-Codierung programmiert.

Jahrgang 10

Die Schülerinnen und Schüler beginnen das Halbjahr mit dem dem Thema **Internet und Netzwerke**. Die Lernenden sollen zunächst eine Zeichnung des Weges eines Social Media Posts von einer Party in Osnabrück zu einem Freund in den USA visualisieren. Nachdem die Grundvorstellungen der Lernenden diskutiert wurden, schauen sie sich einen kurzen Film an, in dem der Weg von Daten im Internet, sowie relevante Netzwerkkomponenten erklärt werden. Die Lernenden sollen anschließend die einzelnen Komponenten beschreiben können und relevante Dienste im Internet identifizieren.

Jetzt, da Grundkenntnisse vorhanden sein sollten, sollen die Lernenden erneut für vier realistische Szenarien die Wege der Daten im Internet auf einem Plakat visualisieren. Hierbei wird auch diskutiert, welche Topologie sich für die Router im Internet anbietet. Abschließend wird in der Einheit ein lokales Heimnetzwerk betrachtet und die Lernenden sollen für einen Ausschnitt des Schulnetzwerkes grafisch darstellen, welches Gerät mit welchen anderen verbunden ist.

Die zweite Einheit des Jahrgangs 10 erweitert die bisherigen Fähigkeiten zum **algorithmischen Problemlösen** mit einer grafischen Programmiersprache und dem Calliope. Nach einer kurzen Wiederholung der bisher erlernten Inhalte werden Variablen mit dem Calliope genutzt und schließen eigene Operationen (Blöcke) an.

Abschließend folgt die dritte und letzte Einheit des Jahrgangs 10 zur **Verwaltung und dem Strukturieren von Daten**. Zunächst entwickeln die Lernenden eigene Ideen, Daten im Kontext "Clubs für Abipartys" zu verwalten. Anschließend wird der Kontext im Szenario Clubbesitzer betrachtet, um grundlegende Strategien zur Modellierung einer Datenbank mit mehreren Tabellen kennenzulernen. Wenn den Lernenden dann bekannt ist, wie Daten in mehreren Tabellen clever strukturiert werden können, werden grundlegende Begriffe zum Thema Datenbanken (u. a. Attribut, Datensatz, Schlüsselattribut, Primärschlüssel, Anomalien,...) eingeführt. Die Einheit endet mit einer Einführung in einfache SQL-Abfragen, welche eigenständig und mit konkreten Datenbanken online durchgeführt wird.

Einführungsphase

Während einer kurzen Wiederholung der bekannten Inhalte der **Codierungstheorie** werden anhand des Vergleichs der Morse-, der ASCII- und eines fehlererkennenden Codes Kriterien einer "guten" Codierung herausgearbeitet. Anschließend werden in mehreren Lernstationen weitere grundlegende Codierungen erarbeitet (Laufängencodierung bei Graubildern, Prüfwerten, Codebäume, Laufängencodierung bei Farbbildern, Entropiecodierungen und eine Vertiefung der RGB- / CMYK-Codierungen). Der Übergang zur Einführung in Algorithmen mit Processing wird anhand des n-bit-Repetitionscode gemacht.

Die Schülerinnen und Schüler lernen beim Thema **Algorithmen** eine textbasierte Programmiersprache Java in der Umgebung Processing im Rahmen eines Lernhelferprojektes, ein Vokabeltrainer, kennen. Nach der Einführung in die Umgebung mit dem Repetitionscode folgt eine Einführung in primitive Datentypen (int, char, String, boolean) und den Umgang mit den bekannten Kontrollstrukturen. Nachdem zusätzlich die Interaktion einer Userin / eines Users eingeführt worden sind, können die Lernenden einen eigenen Vokabeltrainer implementieren. Anhand der allgemeinen Darstellungsformen von Algorithmen in Form von Struktogrammen werden verschiedene Algorithmen anhand von Trace-Tabellen, welche die Belegung von Variablen an verschiedenen Stellen eines Algorithmus verfolgen, analysiert.

Das Thema **Netzwerke** wird mithilfe der Anwendung WebNetSim unterrichtet. Damit können virtuelle Netzwerke mit allen relevanten Einstellungen angelegt werden. Die Wege der Daten in einem häuslichen Netzwerk inklusive der Einstellungen der beteiligten Hardware werden hier ebenso thematisiert, wie die Lagerung von Websites im Internet und deren Aufruf, sowie Mailserver. Der Weg, den Daten im Internet nehmen, wird hier auch thematisiert. Weiterhin werden lebensweltnahe Dienste diskutiert, welche die Lernenden häufig nutzen. Diese sind z. B. Streaming-Dienste, Messenger-Dienste oder Gaming-Dienste.

Das Recht auf informationelle Selbstbestimmung und elementare Regeln im Umgang mit den eigenen Daten im Rahmen von Social Media wird ebenfalls Unterrichtsinhalt sein.

Fächerübergreifend wird hier der Inhalt Datensicherheit und Datenschutz behandelt. Diese Themen berühren durch entsprechende Diskussionen und Gruppenarbeiten mit Präsentationen die Fächer Politik / Wirtschaft, Werte und Normen und auch Philosophie.

Als nächstes Themengebiet wird **Datenanalyse** unterrichtet. Nach einer Problematisierung eines Beispiels im Kontext, welche Daten Unternehmen ihrer Kunden sammeln und was sie damit anfangen können, wird der Unterschied zwischen Korrelation und Kausalität aufgezeigt. Anschließend findet eine Einführung in den Prozess der Datenanalyse, zunächst händisch mit wenigen Daten, statt. Wenn die Grundlagen, inklusive eines Kausalitätsbaums, bekannt sind, wird zur automatisierten Datenanalyse gewechselt. Die Motivation hierbei ist, von einer sehr großen Menge von Daten fiktiver Schüler, deren Familien und den Leistungen dieser Schüler in den ersten beiden Klassenarbeiten, die zu erwartete Leistung der dritten Klassenarbeit vorausszusagen. Hierbei werden die Schüler vom Programm Orange unterstützt, welches detailliert und mit grafischer Unterstützung eine automatisierte Datenanalyse durchführt und jede Menge Interpretationen bzgl. der Qualität der Vorhersagen zulässt. Zum Schluss führen die Schüler eine qualifizierte Beurteilung der Chancen und Risiken der automatisierten Datenanalyse in Form einer arbeitsteiligen Gruppenarbeit durch.

Bereits in der Antike hat sich der Mensch mit der Verschlüsselung von Nachrichten beschäftigt. Wer will schon, dass ein Dritter alles lesen kann, was man versendet?

Im Rahmen der E-Phase geht es in der **Kryptologie** um klassische Verschlüsselungsverfahren, wie die Caesar-Verschlüsselung und Skytale und deren Sicherheit, was die Kryptoanalyse angeht. Die Caesar-Verschlüsselung wird dann auch algorithmisch mit der vorher kennengelernten Programmierungsumgebung umgesetzt. Hierbei werden alle elementaren Kontrollstrukturen und der Umgang mit Variablen und Zeichenketten gefestigt. Außerdem werden andere klassische Verschlüsselungs-

verfahren, bei denen die Zeichen im Klartext nur untereinander vertauscht werden, Transpositionsverfahren, betrachtet und implementiert.

Qualifikationsphase

Das **erste Halbjahr** startet mit einer Beurteilung zur Klassierung von Daten und der Modellierung von Klassen und deren Zusammenhängen mit der unified modelling language (UML). Dadurch wird es ermöglicht, größere Problemstellungen in kleinere Teile zu zerstückeln, um bei der Problemlösung arbeitsteilig vorgehen zu können. Bei korrekter Anfertigung, können aus den Klassendiagrammen alle Klassen unabhängig voneinander implementiert werden. Die Modellierung mit UML ist unabhängig von der konkreten Programmiersprache. Zur Erstellung dieser Diagramme wird die Software yEd-Graphen Editor genutzt und ein Kennenlernen agiler Methoden zur Problemlösung wird eingeführt.

Im Anschluss an die Modellierungseinheit folgt ein Programmierprojekt mit Java, in dem die Lernenden für ein Adventurspielszenario Klassen und Objekte erstellen, und die Klassen um verschiedene Operationen erweitern. In diesen Operationen werden unter anderem Zeichenkettenoperationen, Überladen, Überschreiben und vertiefte gezielte Anwendungen der Kontrollstrukturen verwendet.

Es folgen zwei Sequenzen zu ein- und zweidimensionalen Reihungen. Diese werden im Kontext eines automatisierten Feuerwerks genutzt.

Anschließend wird mit einem Industrieroboterszenarios zu Sortieralgorithmen übergeleitet. Hierfür entwickeln die Lernenden zunächst eigene Sortieralgorithmen und versuchen diese möglichst präzise zu algorithmisieren. Es folgt eine Untersuchung verschiedener Sortieralgorithmen, inklusive dem Erstellen von Struktogrammen und der algorithmischen Umsetzung mit Reihungen und dynamischen Reihungen, wobei letzteres nach den ADT aufgegriffen wird.

Bisher wurde mit Ganzzahlen, Zeichenketten, Dezimalzahlen und Wahrheitswerten mit sogenannten einfachen Datentypen gearbeitet. Oft benötigt man jedoch auch Datenstrukturen, die bestimmte Funktionsweisen haben sollen und dynamisch erweiterbar / kürzbar sein sollen. In der Informatik gibt es die abstrakten Datentypen (ADT Stapel, Schlange, Dynamische Reihung). Deren Umsetzung mit sogenannten Verweisen wird sowohl graphisch, als auch algorithmisch behandelt. Alle ADT werden in verschiedenen Kontexten (Containerstapel, Flughafen, Arztpraxis, Biohof, Mariokart) eingeführt und in diesen gezielt genutzt.

Im Rahmen des ADT Binärbaum lernen die Schülerinnen und Schüler auch Algorithmen kennen, die sich während der Ausführung selbst nutzen. Diese werden als rekursive Algorithmen bezeichnet.

Abschließend folgt eine kleine Sequenz zur Untersuchung von Laufzeiten von Algorithmen.

Im **zweiten Halbjahr** starten die Schüler mit Kompressionsverfahren. Eines der Ziele der Codierungstheorie ist ein möglichst kurzer Code. Hierfür untersuchen die Schüler Bild- und Textkompressionsverfahren, wie die Lauflängen- und die Huffmankodierung. Das zweite Ziel der Codierungstheorie, die Fehlererkennung und Korrektur, folgt anschließend. Hierfür werden erst einfache Repetitions-codes betrachtet und zum Abschluss eine bzgl. Fehlererkennung und Korrektur, wesentlich bessere Codierung, der (7,4)-Hamming-Code.

Den zweite Teil Halbjahres bildet die Kryptologie. Nachdem bereits in der E-Phase einfache Verfahren, wie die Caesar-Verschlüsselung, untersucht wurden, wird die Vigenère-Chiffre als ein Beispiel für polyalphabetische Verschlüsselungen betrachtet.

Um ein Beispiel einer modernene Verschlüsselung kennengelernt zu haben, wird im Anschluss an Vigenère eine vereinfachte Variante des Data Encryption Standard (DES), eine Blockchiffre behandelt.

Abschließend zum Themenkomplex Kryptologie wird sich dem Thema Schlüsseltausch genähert. Bei den bisher untersuchten Verschlüsselungsverfahren gibt es für Sender und Empfänger einen

gemeinsamen Schlüssel, welcher vor der eigentlichen Kommunikation ausgetauscht werden muss. Dies stellt natürlich eine unschöne Angriffsstelle dar, die es zu beseitigen gilt.

Das Verfahren von Diffie-Hellman-(Merkle), ist das hierfür historisch erste veröffentlichte Verfahren, welches dieses Problem beseitigt. Dieser Schlüsseltausch ermöglicht eine asymmetrische Verschlüsselung. Es gibt einen privaten Schlüssel für den Sender und einen anderen für den Empfänger, sowie einen allgemein bekannten öffentlichen Schlüssel. Beim Austausch der Schlüsselinformationen ist es mit diesem Verfahren nicht mehr möglich, die privaten Informationen zu erhalten, da diese nie gesendet werden. Beispielhaft wird eine einfache Hashfunktion implementiert. Weiterhin werden digitale Signaturen und Zertifikate thematisiert.

Die Berechnungen, die für die Verschlüsselungen und den Schlüsseltausch notwendig sind, stellen **fächerübergreifend** einen Bezug zu Mathematik dar. Die Arbeits- und Präsentationsphasen zur Datensicherheit und zum Datenschutz zu Politik-Wirtschaft, Werte und Normen und Philosophie. Inhalte zur globalisierten Welt der Daten spiegeln hinsichtlich dieser Thematik ebenso Inhalte der Geographie wieder.

Der zweite Teil des zweiten Halbjahres hat Datenbanken zum Inhalt.

Datenbanken. Das klingt auf jeden Fall sehr unspannend aber ist für die Schülerinnen und Schüler meist tatsächlich ein beliebtes Thema. Wie muss eine Tabelle oder mehrere Tabellen angelegt sein, damit die Daten darin möglichst effektiv und sinnvoll verwaltet werden können? Diese Fragen werden zur Wiederholung der bekannten Inhalte geklärt. Anschließend werden komplexere SQL-Abfragen eigenständig in Form einer Projektarbeit erarbeitet. Dies geschieht wieder an konkreten Datenbanken online.

Der nächste Inhalt im Bereich Datenbanken ist die Modellierung mit den Entity-Relationship-Diagrammen (ER-Diagramme). Diese werden zum Konzipieren von größeren Datenbanken genutzt, analog zu Klassendiagrammen bei der Programmierung. Die Schülerinnen und Schüler sollen am Ende aus einem Text im Sachzusammenhang, in denen Anforderungen an eine zu erstellende Datenbank stehen, ein solches ER-Modell zu entwickeln. Ebenso sollen sie gegebene ER-Modelle zu Datenbanken umsetzen und aus Datenbanken ER-Modelle aufstellen können. Die Diagramme werden mit dem yED-Graphen-Editor angefertigt und auch hier wird wieder mit agilen Methoden vorgegangen.

Optional kann eine Optimierung von Datenbanken mit Hilfe der ersten drei Normalformen stattfinden.

Im **dritten Halbjahr** ist die theoretische Informatik das erste Thema. Die Funktionsweise von z.B. Futterautomaten ist ein gutes Beispiel zur Theorie der endlichen Automaten. Hinter diesen steckt auch immer eine sogenannte formale Sprache, mit welcher Wörter gebildet werden, die der Automat als Eingabe akzeptiert.

Die Schülerinnen und Schüler entwickeln hier aus verschiedenen Sachzusammenhängen Automaten und auch die entsprechenden formalen Sprachen. Am Ende sollen sie in der Lage sein, zu einer regulären Sprache den Zustandsgraph eines solchen Automaten zu erstellen, aus einem solchen Graph die Sprache abzuleiten und gegebene Zustandsgraphen zu minimieren. Gearbeitet wird hier viel mit dem Programm AtoCC bzw. der Onlinevariante Flaci.

Grenzen von endlichen Automaten werden ebenfalls untersucht und Kellerautomaten, sowie optional NEA, eingeführt. Der Bereich der formalen Sprachen ist bezüglich Grammatiken, Bildungsregeln und Ableitungsbäumen für jede Sprache des Aufgabenfeldes A **fächerübergreifend** relevant.

Das letzte Thema des dritten Halbjahres ist künstliche Intelligenz. Nachdem mit Hilfe des Online-spiels Akinator, welcher nach wenigen Fragen recht zielsicher eine vorher geheim überlegte Person errät, ein motivierender Einstieg ins Thema gefunden wurde, werden zunächst Fachbegriffe und Grundlagen zu KI geklärt. Anschließend findet eine Erläuterung zum maschinellen Lernen statt, nach welchem die Schüler an Beispielen wie Lebensmittel kategorisieren und einer Gesichtserkennung bzw. -klassifizierung (Mann/Frau) diesen Lernprozess durchführen und nachempfin-

den können. Dies geschieht mit Hilfe von Entscheidungsbäumen. Didaktisch reduziert sollen die Lernenden die hierfür notwendigen GINI-Scores in einer eigens zu erstellenden Tabellenkalkulation automatisch errechnen lassen. Das Erstellen der Bäume kann anschließend selbst durchgeführt werden. Zum Abschluss findet eine Diskussion zu "Kann ein Algorithmus rassistisch sein?" statt, welcher zuvor mit realen Beispielen hierzu eingeleitet wird. Als optionale Ergänzung oder im erhöhten Anforderungsniveau schließt noch eine weitere Sequenz zu Reinforcement Learning an, wobei die Schüler hier selbst anhand dreier Spiele in Gruppen den Lernprozess einer Maschine nachempfinden und anschließend analysieren.

Das **vierte Halbjahr** kann weitestgehend optional gestaltet werden. Es gibt verschiedene Vorschläge.

Einmal können vorangegangene Inhalte wiederholt und geübt werden.

Ein weiterer Vorschlag ist die technischen Informatik. Hier wird praktisch gearbeitet. Nach einer kurzen Einführung in logische Ausdrücke und dem Umgang mit diesen, werden auf physischer Ebene Schaltungen gebaut. Die Schülerinnen und Schüler lernen hier mit Hilfe elementarer Logikgatter und den entsprechenden Bauteilen Schaltungen zu bauen, welche auch im Sachzusammenhang bestimmte Probleme lösen. Schaltterme werden systematisch hergeleitet und minimiert und daraus werden Schaltnetze gebaut.

Als Vorstufe wird die Software LogiSim oder der Digital Simulator online genutzt, mit denen virtuelle Schaltungen gebaut und getestet werden können. Im Anschluss werden diese auch auf Steckplatinen zusammengebaut. Hier kann auch gut arbeitsteilig vorgegangen werden, sodass größere Schaltungen von mehreren Gruppen in Teilen erstellt und anschließend zusammengefügt werden. Es wird auch thematisiert, inwiefern die Automaten mit Ausgabe aus der vorherigen Einheit physisch umgesetzt werden können. Da in diesem Teil des dritten Halbjahres verstärkt Berechnungen, auch komplexerer Art, auftauchen, ist ein **Fächerübergreif** zu Mathematik festzustellen.

Verschiedene Vertiefungen der Themen der ersten drei Halbjahre können hier ebenfalls durchgeführt werden. Alternativ ist auch eine Projektarbeit denkbar, in welcher z. B. ein Produkt am Ende des Halbjahres entstanden ist. Dies kann ein Programm sein, aber auch ein physisches Ergebnis, welches man in der Schule ausstellen kann.

Bezug zum Medienbildungskonzept

Medienkompetenz, wie sie der Informatikunterricht in besonderer Weise vermittelt, hat damit in der heutigen Zeit eine ähnliche Bedeutung wie Lesen, Schreiben und Rechnen. Sie ist Voraussetzung für die Teilhabe an der Informations- und Wissensgesellschaft sowie an demokratischen Prozessen der Meinungsbildung. In allen Berufen und Wissenschaften haben sich die Arbeitsweisen durch die Methoden der Informatik und den Einsatz informatischer Systeme grundlegend verändert. Dies erfordert ein vertieftes Verständnis informatischer Zusammenhänge, um über das reine Anwenden hinaus, kreativ, reflektiert und zielgerichtet mit Informatiksystemen arbeiten zu können. Die dazu notwendigen Verfahren und Kenntnisse werden insbesondere durch den Informatikunterricht vermittelt.

Präsentationen werden im Informatikunterricht am Graf-Stauffenberg-Gymnasium generell elektronisch unterstützt gehalten. Die Bearbeitung der einzelnen Aufgaben involviert in allen Themen spezielle Softwarenutzung. Diese sind unter anderem yEd, Processing, LogiSim, Textverarbeitungssoftware, Präsentationssoftware, Structorizer, AtoCC, Scratch, Scratch, Filius und verschiedene Onlineprogrammen. Im Bereich der technischen Informatik werden auf Steckplatinen mit LED, Kabeln und Logikgattern einfache und komplexe Schaltungen physisch aufgebaut.

Hinsichtlich des Orientierungsrahmens Medienbildung werden die dort vorgeschlagenen Kompetenzen in den Inhalten zu den einzelnen Jahrgängen entsprechenden Unterrichtsinhalten zugeordnet. Die fünf Kategorien sind in den folgenden Tabellen abgebildet. Bei den Inhalten der Jahrgänge werden diese zum Beispiel als OM 1.4.2 bezeichnet, wenn aus dem Kompetenzbereich "1. Suchen, Erheben, Verarbeiten und Aufbewahren" die vierte Zeile auf der Kompetenzstufe 2, also "...analysieren und strukturieren themenrelevante Informationen aus Medienangeboten" gemeint ist.

Die Schülerinnen und Schüler des Graf-Stauffenberg-Gymnasiums dokumentieren ihren Lernprozess und erstellen sämtliche Produkte digital, mit Hilfe verschiedenster Programme und Internetseiten. Deshalb sind die Kompetenzen 3.1-4 auf allen Kompetenzstufen dauerhaft in die Unterrichtsdurchführung integriert.

Das Arbeiten im Informatikunterricht ist durchgehend durch einem PC begleitet und auf diesem werden Textdokumente und Präsentationen erstellt, programmiert und eine Vielfalt an Software und Internetseiten genutzt. Für Details sei auf die Hinweise zur inhaltlichen Umsetzung verwiesen. Ein Tablet ist aufgrund der geringen Displaygröße und bei Fehlen von Maus und Tastatur als Arbeitswerkzeug für den Informatikunterricht ungeeignet, wenn man von der Nutzung als Lesegerät oder als Schreibblockersatz absieht.

1. Suchen, Erheben, Verarbeiten und Aufbewahren		
Kompetenzstufe 1	Kompetenzstufe 2	Kompetenzstufe 3
	Die Schülerinnen und Schüler...	
... informieren sich unter Anleitung mithilfe von Medien.	... recherchieren in verschiedenen digitalen Umgebungen.	... führen selbständig komplexe Medienrecherchen durch.
... beherrschen den Umgang mit altersgerechten Suchmaschinen.	... analysieren relevante Quellen. ... verarbeiten Suchergebnisse.	
... entwickeln erste Suchstrategien.	... analysieren und strukturieren themenrelevante Informationen aus Medienangeboten.	... interpretieren Informationen aus Medienangeboten und bewerten diese kritisch.
... entnehmen zielgerichtet Informationen aus altersgerechten Informationsquellen.	... organisieren, strukturieren und sichern Daten und Informationen.	... bereiten Informationen unter Berücksichtigung rechtlicher Vorgaben auf.
... erarbeiten Kriterien zur Bewertung von Informationsquellen.	... analysieren und vergleichen Inhalt, Struktur, Darstellungsart und Zielrichtung von Daten- und Informationsquellen.	... bewerten kriteriengeleitet Informationsquellen und Suchergebnisse.
... speichern Daten und Informationen sicher und auffindbar.	... rufen Daten und Informationen von verschiedenen Orten ab.	... sichern Ergebnisse mit selbstgewählten Methoden und Strategien.

2. Kommunizieren und Kooperieren		
Kompetenzstufe 1	Kompetenzstufe 2	Kompetenzstufe 3
	Die Schülerinnen und Schüler...	
... kommunizieren und interagieren mit Hilfe verschiedener digitaler Kommunikationsmöglichkeiten.	... passen ihre Kommunikation und ihr Verhalten der jeweiligen digitalen Umgebung an.	... verwenden verschiedene digitale Kommunikationsmöglichkeiten zielgerichtet, adressaten- und situationsgerecht.
... sprechen über Medienerfahrungen und tauschen sich über Wahrnehmung und Wirkung von medialer Kommunikation aus.	... geben Erkenntnisse aus Medienerfahrungen weiter und bringen diese in kommunikative und kooperative Prozesse ein.	... beteiligen sich an gesellschaftlichen Diskursen, und nutzen ihre Medienerfahrungen und Kommunikationsmöglichkeiten zur aktiven gesellschaftlichen Partizipation.
... formulieren Verhaltensregeln für digitale Umgebungen und wenden diese an.	... beurteilen Chancen und Risiken verschiedener Kommunikationsmöglichkeiten in digitalen Umgebungen. ... reflektieren die Nutzung digitaler Kommunikationsmöglichkeiten und kommunizieren verantwortungsbewusst.	... berücksichtigen ethische Prinzipien und kulturelle Vielfalt bei der Kommunikation und Kooperation in digitalen Umgebungen.
... nutzen in kooperativen Arbeitsprozessen digitale Werkzeuge.	... führen in kooperativen Arbeitsprozessen mit digitalen Werkzeugen Daten, Informationen und Ressourcen zusammen.	... kooperieren selbständig, reflektiert sowie verantwortungsbewusst in digitalen Umgebungen.
... teilen Dateien, Informationen und Links (inkl. Quellenangabe).	... wählen in kooperativen Arbeitsprozessen passende digitale Werkzeuge bzw. Plattformen zum Teilen und beherrschen die Referenzierungspraxis (Quellenangaben).	... teilen ihre Produkte unter Berücksichtigung von Urheber- und Nutzungsrecht.

3. Produzieren und Präsentieren		
Kompetenzstufe 1	Kompetenzstufe 2	Kompetenzstufe 3
	Die Schülerinnen und Schüler...	
... sammeln und führen vorhandene Inhalte in verschiedenen Formaten zusammen.	... können technische Bearbeitungswerkzeuge sowie ästhetische Gestaltungsmittel benennen und wenden diese in verschiedenen Formaten (Text, Bild, Audio, Video, etc.) an.	... setzen unterschiedliche Gestaltungsmittel zielgerichtet ein und dokumentieren ihren Produktionsprozess.
... planen und entwickeln unter Anleitung Medienprodukte in vorgegebenen Formaten.	... verarbeiten Informationen, Inhalte und vorhandene digitale Produkte weiter.	... integrieren eigene digitale Produkte in bestehendes Wissen.
... präsentieren, veröffentlichen oder teilen ihre Medienprodukte.	... präsentieren ihre Medienprodukte unter Einsatz digitaler Werkzeuge. ... definieren Kriterien für die Beurteilung von Medienprodukten.	... wählen geeignete Präsentationsformen für eine sach- und adressatengerechte Veröffentlichung ihrer Medienprodukte. ... geben kriteriengeleitet Rückmeldung zum Medienprodukt und zur Präsentation.
... beachten den Unterschied zwischen eigenen und fremden Medienproduktionen.	... kennen die Bedeutung von geistigem Eigentum.	... berücksichtigen Persönlichkeits-, Urheber- und Nutzungsrechte bei ihren Gestaltungs- und Produktionsprozessen.

4. Schützen und sicher Agieren		
Kompetenzstufe 1	Kompetenzstufe 2	Kompetenzstufe 3
	Die Schülerinnen und Schüler...	
... benennen Risiken und Gefahren digitaler Umgebungen und entwickeln ein grundlegendes Sicherheitsbewusstsein.	... reflektieren ihr eigenes Nutzungsverhalten und vereinbaren Regeln zum Umgang mit digitalen Endgeräten.	... agieren sicher und verantwortungsbewusst in digitalen Umgebungen.
... wenden einfache Strategien zum Merken und Verwalten sicherer Zugangsdaten an (zum Beispiel personalisierte Accounts).	... entwickeln ein Bewusstsein für Datensicherheit, Datenschutz und Datenmissbrauch, um ihre Privatsphäre durch geeignete Maßnahmen zu schützen.	... entwickeln ein Bewusstsein für ethische Fragen der Datenverwendung und bringen ihre Haltung in den gesellschaftlichen Diskurs ein.
... nutzen Strategien zur Herstellung von Gerätesicherheit (Virens Scanner, Firewall, Zugangssicherung etc.).	... wenden Strategien zum Selbstschutz und zur Selbstkontrolle an.	... nutzen digitale Technologien für soziales Wohlergehen und Eingliederung.
	... reflektieren ihre Selbstdarstellung in sozialen Medien.	... nutzen bewusst Internetdienste und wenden dort Strategien zum Schutz personenbezogener Daten an.
... benennen die Chancen und Risiken (finanziell, sozial, gesundheitlich) digitaler Umgebungen.	... nutzen digitale Werkzeuge (z. B. Apps) zur Förderung der eigenen Gesundheit.	... identifizieren Mechanismen zur Verbraucher manipulation und setzen mögliche Schutzmaßnahmen ein.
	... berücksichtigen Jugendschutz und Verbraucherschutzmaßnahmen.	
... beschreiben die Auswirkungen digitaler Technologien auf Natur und Umwelt.	... hinterfragen ihr eigenes Nutzungsverhalten digitaler Technologien hinsichtlich der Auswirkungen auf Natur und Umwelt.	... nutzen digitale Technologien zum Schutz von Natur und Umwelt.

5. Problemlösen und Handeln		
Kompetenzstufe 1	Kompetenzstufe 2	Kompetenzstufe 3
	Die Schülerinnen und Schüler...	
... beachten die Unterschiede verschiedener digitaler Umgebungen (z. B. Apps, Webseiten, Textverarbeitung, Tabellenkalkulation).	... formulieren Anforderungen an digitale Lernumgebungen.	... bewerten und nutzen effektive digitale Lernmöglichkeiten und digitale Werkzeuge sowie Medien zum Lernen, Arbeiten und Problemlösen.
... benennen einfache Grundfunktionen von digitalen Werkzeugen (zur Verarbeitung von Daten und Informationen).	... setzen Werkzeuge bedarfsgerecht ein.	... beurteilen verwendete digitale Werkzeuge im Hinblick auf den Datenschutz und mögliche gesellschaftliche Auswirkungen.
... wählen altersgemäße digitale Lernmöglichkeiten aus.	... beurteilen den Nutzen digitaler Lernumgebungen für den persönlichen Gebrauch.	... passen digitale Umgebungen und Werkzeuge zum persönlichen Gebrauch an.
... wenden einfache Funktionen von digitalen Werkzeugen (unter Anleitung) an.	... erarbeiten und formulieren erste algorithmische Zusammenhänge.	... erkennen und formulieren algorithmische Strukturen in digitalen Werkzeugen.
		... planen und verwenden eine strukturierte, algorithmische Sequenz bei der Erstellung eigener Programme zur Problemlösung.
... identifizieren technische Probleme.	... ermitteln Bedarfe für die Lösung	... finden Lösungen für technische Probleme und verstehen Funktionsweisen sowie grundlegende Prinzipien der digitalen Welt.
	... identifizieren passende Werkzeuge zur Problemlösung.	
	... entwickeln Lösungsstrategien technischer Probleme.	
... erkennen eigene Defizite bei der Nutzung digitaler Werkzeuge.	... entwickeln und nutzen Strategien zur Beseitigung eigener Defizite bei der Nutzung digitaler Werkzeuge.	... setzen Lösungsstrategien effektiv um.
		... teilen Strategien zur Beseitigung eigener Defizite bei der Nutzung digitaler Werkzeuge mit anderen.

6. Analysieren, Kontextualisieren und Reflektieren		
Kompetenzstufe 1	Kompetenzstufe 2	Kompetenzstufe 3
	Die Schülerinnen und Schüler...	
... stellen die Vielfalt der altersgemäßen Medienangebote und ihre Zielsetzungen dar.	... reflektieren die Wirkung von Medien auf das eigene Handeln sowie auf Individuum und Gesellschaft.	... beurteilen durch Medien vermittelte Rollen- und Wirklichkeitsvorstellungen.
... erkennen die Wirkung unterschiedlicher Darstellungsmittel in und durch Medien.	... analysieren und vergleichen Medienangebote und deren Gestaltungsmittel.	... bewerten Medienangebote und deren Gestaltungsmittel. ... reflektieren die Bedeutung von Medien für die Identitätsbildung (auch in gesellschaftlichen Kontexten).
... beschreiben ihr eigenes Medienverhalten.	... erkennen die Folgen ihrer Mediennutzung auf ihr persönliches Umfeld.	... setzen sich mit dem Problem der ständigen Verfüg- und Erreichbarkeit kritisch auseinander.
... setzen sich mit ihrem eigenen Medienverhalten auseinander.	... analysieren und modifizieren den eigenen Mediengebrauch. ... orientieren ihre Mediennutzung an (z. B. durch das Grundgesetz formulierten) Werten.	... analysieren und reflektieren den Einfluss von Medien auf gesellschaftliche Prozesse und Werte. ... reflektieren die Potentiale der Digitalisierung im Sinne sozialer Integration sowie zur gesellschaftlichen Partizipation.
...verarbeiten ihre Medienerlebnisse, indem sie Regeln für den bewussten Umgang mit Medien entwickeln.	... schätzen die Auswirkungen digitaler Technologien auf wirtschaftliche, soziale und politische Prozesse ein.	... beurteilen die Bedeutung medialer Darstellungen für die Bewusstseinsbildung und die politische Meinungsbildung. ... beurteilen die Entwicklung digitaler Medien und Technologien.

Konkrete Medienbildungsinhalte der einzelnen Jahrgänge

Jahrgang 9		
Inhalt	Zum Thema	Konkrete Umsetzung
Algorithmisches Problemlösen	Einführung in das Algorithmische Problemlösen	Die SuS entdecken, was ein Algorithmus ist und werden an das Lösen von verschiedenen Problemen in einer grafischen Programmiersprache herangeführt.
EVA-Prinzip	Grundlagen der Datenverarbeitung	Die SuS lernen in einem Planspiel, welche Komponenten für die Eingabe, die Verarbeitung eines Algorithmus und die Ausgabe in einem Computer bzw. Informationssystem zuständig sind.
Daten und ihre Darstellung im digitalen Kontext	Grundlagen der Datenverarbeitung	Die SuS werden in die Notwendigkeit digitaler Codierungen von Daten herangeführt, entdecken verschiedene gängige Codierungen und wenden diese an.
Basiskompetenzen	Alle	Ordnerstruktur, Gezieltes Speichern und Laden von Dateien, Allgemeiner Umgang mit dem Computer, Software und einem Browser

Jahrgang 10		
Inhalt	Zum Thema	Konkrete Umsetzung
Algorithmisches Problemlösen	Algorithmisches Problemlösen	Die SuS vertiefen ihre Kenntnisse und Fähigkeiten zum algorithmischen Problemlösen. Außerdem können sie Algorithmen standardisiert darstellen.
Daten und ihre Darstellung im digitalen Kontext	Verwaltung von Daten	Die SuS werden in die effiziente Verwaltung von Daten eingeführt und können einfache Abfragen an Datenbanken ausführen.
Aufbau und Komponenten des Internets	Aufbau des Internets	In einem Planspiel zum Aufbau des Internets entdecken die SuS die essentiellen Komponenten des Internets (Client, Server, Router, Switch, DNS) und deren Funktionsweise.
Sicherheit durch Passwörter und Verschlüsselung von Daten	Aufbau des Internets	Die SuS recherchieren zu verschiedenen Themen zur Datensicherheit.
Basiskompetenzen	Alle	Ordnerstruktur, Gezieltes Speichern und Laden von Dateien, Allgemeiner Umgang mit dem Computer, Software und einem Browser

Jahrgang 11		
Inhalt	Zum Thema	Konkrete Umsetzung
Textverarbeitungssoftware und Tabellenkalkulation	Schülerlösungen werden mit Libre- oder MS-Office angefertigt	Die SuS fertigen in weiten Teilen ihre Arbeitsergebnisse in digitaler Form an. Dies beinhaltet auch die Nutzung verschiedener Formatierungen und das Einbinden von Tabellen und Bildern in die entsprechenden Dokumente.
Umgang mit dem Internet	Thema Datenschutz: ca. 2-3 Wochen	Die SuS können anhand konkreter Szenarien und allgemein das Aufzeichnen und die Weitergabe personenbezogener Daten hinsichtlich der rechtlichen Rahmenbedingungen und der Bedeutung für die Einzelperson und die Gesellschaft im Allgemeinen einschätzen.
Aufbau von Netzwerken und dem Internet	Thema Datenaustausch in Netzwerken: ca. 6 Wochen	Die SuS erstellen eigene Netzwerke mit den notwendigen Einstellungen zur Kommunikation der beteiligten Geräte untereinander. Ebenfalls werden IP-Adressen automatisiert zugewiesen, sowie DNS-, Web- und Mail-Server und mehrere Netzwerke virtuell aufgebaut.
Funktionsweise von Computern	Visuelle und textbasierte Programmierung: ca. ein Halbjahr	Die SuS können elementare Kontrollstrukturen von Algorithmen zu konkreten Problemlösungen zu einem Algorithmus zusammenstellen. Zusätzlich werden Variablen, elementare Datentypen und eigene Methoden genutzt.
Verschlüsselung von Daten	Thema Kryptologie ca. ein 5 Wochen	Die SuS können die Notwendigkeit von Verschlüsselungen beschreiben und konkrete klassische Verschlüsselungen anwenden. Mit Hilfe von kryptoanalytischen Mitteln können sie erläutern, weshalb klassische Verschlüsselungsverfahren nicht sicher sind.
Informationen digitalisieren	Thema Codierungstheorie: ca. 5 Wochen	Die SuS können die Notwendigkeit Informationen zu codieren beschreiben und anhand verschiedener Codierungen wünschenswerte Eigenschaften einer Codierung erläutern. Weiterhin können die SuS verschiedene Codierungen anwenden und hinsichtlich ihren Eigenschaften analysieren.
Basiskompetenzen	Alle	Ordnerstruktur, Gezieltes Speichern und Laden von Dateien, Allgemeiner Umgang mit dem Computer, Software und einem Browser

Jahrgang 12		
Inhalt	Zum Thema	Konkrete Umsetzung
Modellierung	UML-Klassendiagramme und Datenbanken	Die SuS entdecken Situationen, in denen eine Klassierung von verschiedenen Objekten sinnvoll ist und entwerfen für konkrete Kontexte Klassenkarten und -diagramme. Zum Thema Datenbanken entwerfen die SuS, entsprechend verschiedener Anforderungsbeschreibungen, ER-Modelle für eine Datenbank und beurteilen ihre Entwürfe.
Klassen und Objekte	OOP	Die SuS implementieren verschiedene Klassen mit deren Attributen, Konstruktoren und verschiedenen Operationen in einer textbasierten Programmiersprache. Außerdem lernen die SuS verschiedene primitive Datentypen kennen und nutzen diese.
Optional: Grafische Darstellung	Algorithmen	Die SuS nutzen bekannte ADT und Algorithmen zur Entwicklung von Anwendungen mit einem GUI.
Bedarfsgerechte Erstellung und Nutzung von ADT im Kontext	lineare und nichtlineare ADT	Die SuS erweitern ihre Kenntnisse zu Datentypen mit in verschiedenen Kontexten notwendigen, teils abstrakten, Datentypen (Reihung, zweidimensionale Reihung, Stapel, Schlange, dynamische Reihung, Baum und Binärbaum) und wenden diese in einer textbasierten Programmiersprache zur Lösung verschiedener Probleme an.
Daten sortieren	Algorithmen	Die SuS entdecken verschiedene Sortierverfahren, algorithmisieren diese, zunächst in Wortform und Struktogramm, anschließend in einer textbasierten Programmiersprache und wenden sie in verschiedenen Situationen an.
Optimierung von Algorithmen	Laufzeituntersuchungen	Die SuS vergleichen mit Hilfe von Zählvariablen für Vergleiche und Speicherungen verschiedene Implementationen von Algorithmen hinsichtlich ihrer Effizienz.
Informationen situationsabhängig darstellen	Codierungstheorie	Die SuS vertiefen ihre Kenntnisse aus den Jahrgängen 10 und 11 zur Codierungstheorie, indem sie zunächst Kompressionsverfahren und danach fehlererkennende und -korrigierende Codierungen kennenlernen, anwenden und in Teilen implementieren.
Sicherheit bei der Datenübertragung	Kryptologie	Die SuS vertiefen Ihre Kenntnisse zu Verschlüsselungsverfahren und Kryptoanalyse, indem sie unter anderem eine polyalphabetische Verschlüsselung, DES und asymmetrische Verfahren kennenlernen, nutzen und in Teilen implementieren.
Anforderungen an eine Datenübertragung	Protokolle	Die SuS entdecken, welche Absprachen zwischen Kommunikationspartnern notwendig sind und setzen diese in Form eines Protokolls zur Datenübertragung um.
Strukturierung und Nutzung von Daten	Datenbanken	Die SuS lernen grundlegende Notwendigkeiten für das Anlegen und Verwalten größerer Datenmengen in Datenbanken kennen, beurteilen gegebene Datenbanken und erweitern sie gegebenenfalls und wenden SQL-Abfragen auf Datenbanken an.
Basiskompetenzen	Alle	Ordnerstruktur, Gezieltes Speichern und Laden von Dateien, Allgemeiner Umgang mit dem Computer, Software und einem Browser

Jahrgang 13		
Inhalt	Zum Thema	Konkrete Umsetzung
Automaten als Akzeptoren und Automaten mit Ausgabe	Theoretische Informatik	Die SuS nutzen deterministisch endliche Automaten als Akzeptoren und Mealy-Automaten als Automaten mit Ausgabe, um verschiedene Probleme im Kontext zu lösen. Weiterhin können die SuS Situationen, welche endliche Automaten nicht mehr lösen können, erkennen und erläutern.
Optional: Turingmaschinen	Theoretische Informatik	Die SuS lernen Turingmaschinen und entsprechende Anwendungen kennen.
Chancen und Risiken automatisierter Datenanalyse	Datenanalyse	Nach einer Einführung zum Unterschied zwischen Kausalität und Korrelation entdecken die SuS Vorgehensweisen bei einer manuellen Datenanalyse, erweitern ihre Kenntnisse zur automatisierten Datenanalyse mit einer entsprechenden Software und beurteilen ihre Auswertungen. Anschließend sind sie nach Untersuchung verschiedener realer Szenarien, bei denen automatisierte Datenanalyse zum Einsatz kommt, in der Lage die Chancen und Risiken dieser Technik für die Gesellschaft zu beurteilen.
Chancen und Risiken durch KI	KI	Die SuS werden in das maschinelle Lernen und Klassifikationsbäume eingeführt und führen selbst eine Klassifikation durch. Nach einer weiteren Einführung in Auswirkungen von KI in verschiedenen realen Szenarien können sie Chancen und Risiken durch KI in für die Gesellschaft beurteilen.
Technische Umsetzung von Schaltnetzen	Technische Informatik	Die SuS werden in logische Ausdrücke und einfache Logikgatter eingeführt, entwerfen zunächst einfache, später komplexere, Schaltnetze zur technischen Lösung verschiedener Probleme und können Schalterterme systematisch erstellen und auch mit einem Algorithmus minimieren.
Optional: Optimierungsprobleme	Graphen	Die SuS lernen den ADT Graph kennen, entdecken Lösungen der Probleme des kleinsten Weges, der Breiten- und Tiefensuche.
Basiskompetenzen	Alle	Ordnerstruktur, Gezieltes Speichern und Laden von Dateien, Allgemeiner Umgang mit dem Computer, Software und einem Browser

Bezug zum Berufsorientierungskonzept

Das Fach Informatik ermöglicht den Lernenden aller Jahrgänge in verschiedenen Berufsfeldern praktische Erfahrungen zu sammeln und berufsrelevante Kompetenzen zu erwerben.

Die einzelnen Elemente zum Beitrag des Fachs Informatik am Graf-Stauffenberg-Gymnasium Osnabrück für das Berufsorientierungskonzeptes sind im Folgenden für die einzelnen Jahrgänge aufgeführt. Durchweg werden Lösungen und Ergebnisse in allen Jahrgängen mit selbst gewählten Textverarbeitungsprogrammen und Präsentationssoftware erstellt.

Jahrgang 5		
Inhalt	Zum Thema	Konkrete Umsetzung
Allgemeines zu Computern	Einführung	Die Lernenden können mit ein Betriebssystem und IServ grundlegend nutzen und Emailstandards nennen und anwenden.
Dateien	Dateiverwaltung und -formate	Die Lernenden können gängige Dateiformate beschreiben, eine sinnvolle Ordnerstruktur anlegen und anwenden, Dateianhänge von Emails nutzen und Dateien gezielt herunterladen und Speichern.
Textverarbeitung	Textverarbeitung	Die Lernenden können ein Programm zur Textverarbeitung nutzen und grundlegende Formatierungsoptionen erläutern und nutzen.
Tabellenkalkulation	Daten speichern und auswerten	Die Lernenden können ein Programm zur Tabellenkalkulation nutzen, um Daten im Kontext geschickt zu speichern und Informationen (Summe, max, min, Mittelwert, Anzahl und Berechnungen) aus diesen zu gewinnen.
Präsentationen	Präsentationen	Die Lernenden können gewünschte Eigenschaften einer "guten" Präsentation nennen und erläutern und eine solche Präsentation im Kontext mit einer Präsentationssoftware erstellen und vorstellen. Hierbei werden auch grundlegende Gestaltungselemente genutzt.
Suchmaschinen	Suchen im Internet	Die Lernenden können Suchanfragen analysieren und beurteilen, sowie geeignete Suchanfragen erstellen.
Rechtliche Aspekte	Rechtliche Aspekte bei Veröffentlichungen im Internet	Die Lernenden können aufgrund grundlegender, rechtlicher Aspekte Fallbeispiele analysieren und beurteilen, sowie den eigenen Umgang mit Foto-, Video-, Audio- und anderen Dateien entsprechend bewerten.

Jahrgang 9		
Inhalt	Zum Thema	Konkrete Umsetzung
Physical Computing	Einführung in das algorithmische Problemlösen	Die Lernenden nutzen Sensoren und Aktoren, um mit einem Informatiksystem kontextbezogene Probleme zu lösen.
Programmiersprachen	Einführung in das algorithmische Problemlösen	Die Lernenden nutzen eine grafische Programmiersprache und lernen alle relevanten Grundlagen eines Algorithmus kennen.
Computer	Arbeitsweise eines Computers - Hardwarekomponenten	Die Lernenden kennen alle grundlegenden Komponenten eines Computers.
Informationsaustausch	Grundlagen der Datenverarbeitung	Die Lernenden können verschiedene Anforderungen eines Kommunikationsprotokolls, sowohl digital, wie auch analog, entsprechend umsetzen und Daten geeignet codieren.

Jahrgang 10		
Inhalt	Zum Thema	Konkrete Umsetzung
Architektur von vernetzten Systemen	Der Aufbau des Internets	Die Lernenden kennen alle zentralen Komponenten eines Netzwerks und des Internets, können deren Funktionen und den Weg von Daten im Internet erläutern.
Programmiersprachen	Vertiefung zum algorithmischen Problemlösen	Die Lernenden vertiefen ihre Kompetenzen, in einer grafische Programmiersprache Algorithmen zum Lösen eines realen Problems zu entwerfen und zu analysieren. Außerdem können sie Algorithmen standardisiert, in Form von Nassi-Shneidermann-Struktogrammen darstellen.
Datenverwaltung	Einführung in Datenbanken	Die Lernenden können grundlegende Ideen zur effizienten Datenverwaltung in verknüpften Tabellen anwenden und gezielt an Informationen mit einfachen SQL-Anweisungen aus Datenbanken gelangen.

Jahrgang 11		
Inhalt	Zum Thema	Konkrete Umsetzung
Programmiersprachen	Algorithmen und Datenstrukturen	Die Lernenden vertiefen ihre Kompetenzen, Algorithmen zum Lösen eines realen Problems zu entwerfen und zu analysieren, mit einer textbasierten Programmiersprache.
Informationsaustausch	Vertiefung der Codierungstheorie	Die Lernenden vertiefen ihre Kompetenzen zur Codierungstheorie (Kompressionsverfahren, Fehlererkennung und -korrektur, RGB-Codierung)
Architektur von vernetzten Systemen	Datenaustausch in Netzwerken	Die Lernenden vertiefen ihre Kompetenzen zu Netzwerken, indem sie gegebene Netzwerke analysieren und notwendige, fehlerbehebende Einstellungen vornehmen.
Datenanalyse und Datenschutz	Datenanalyse und DSGVO	Die Lernenden analysieren Chancen und Risiken der automatisierten Datenanalyse auf die Gesellschaft und Individuen und können die Grundprinzipien des Datenschutzes (auf Basis der DSGVO) erläutern und Fallbeispiele analysieren.
Daten verschlüsseln	Einführung in die Kryptologie	Die Lernenden können Daten gegen Angreifer mit einfachen, klassischen Verfahren schützen und deren Schwachstellen im Rahmen der Kryptanalyse erläutern, sowie geeignete und sichere Passwörter erstellen.

Jahrgang 12		
Inhalt	Zum Thema	Konkrete Umsetzung
Algorithmische Modellierung und Umsetzung von Problemen im Kontext	Algorithmen, Objektorientierte Modellierung und Programmierung, Abstrakte Datentypen	Die Lernenden können Situationen im Kontext mit geeigneten Datentypen, inklusive linearer und nicht-linearer ADT) und Klassen modellieren und diese in entsprechende Algorithmen und Operationen (textbasiert) umsetzen. Weiterhin können die Lernenden Datenmengen sortieren und auf diesen geeignete Suchen ausführen.
Informationsaustausch	Codierungstheorie	Die Lernenden vertiefen ihre Kompetenzen zur Codierungstheorie (Hamming-Code, Eurobanknoten, QR, ISBN, EAN,...)
Daten verschlüsseln	Kryptologie	Die Lernenden können moderne Verschlüsselungsverfahren (symmetrische und asymmetrische, inklusive digitaler Signaturen) erläutern, anwenden und implementieren.
Datenverwaltung	Datenbanken	Die Lernenden können komplexe Datenbankmodelle zu gegebenen Anforderungen, auch im Rahmen agiler Methoden, entwerfen und zu einer Datenbank umwandeln. Die Kompetenz, mit SQL gezielte Informationen aus Datenbanken zu gewinnen, wird vertieft.

Jahrgang 13		
Inhalt	Zum Thema	Konkrete Umsetzung
Automatisierte Prozessen	Theoretische Informatik	Die Lernenden können mit verschiedenen Automatenmodellen Sachprobleme modellieren und lösen.
Maschinelles Lernen	Künstliche Intelligenz	Die Lernenden können mit Hilfe des Maschinellen Lernens und des Reinforcement Learnings die Funktionsweise künstlicher Intelligenzen erläutern und beide Verfahren anwenden. Weiterhin werden Chancen und Risiken für die Gesellschaft und Individuen, die sich durch KI-Anwendungen ergeben, analysiert und beurteilt.
Konstruktion technischer Systeme	Einführung in die technische Informatik	Die Lernenden können zu Sachproblemen entsprechende Schaltnetze entwerfen und konstruieren. Weiterhin können sie die konkrete Umsetzung eines Automaten in ein Schaltwerk erläutern.
Quantenverschlüsselung	Quantencomputing	Die Lernenden können, nach einer Einführung in die Grundlagen der Quantenmechanik, ein Protokoll für einen asymmetrischen Schlüsseltausch, basierend auf Quantencomputing, erläutern und durchführen.

Sprachsensibler Unterricht

Sprache ist eine notwendige Voraussetzung, um im schulischen Kontext, im späteren Arbeitsleben und gesellschaftlich möglichst gut zu partizipieren. Die Aufgabe eines sprachsensiblen Unterrichts ist allgemein, allen Schülerinnen und Schülern durch bewusste und gezielte sprachliche Unterstützung das fachliche Lernen zu ermöglichen.

Im Fach Informatik stellen sich den Lernenden verschiedene Herausforderungen. Gerade mit Blick auf die Oberstufe und den Anforderungen der Abituraufgaben stellt sich für die Lehrenden die Aufgabe, den Schülerinnen und Schülern zu ermöglichen umfangreiche, komplexe und fachsprachliche Texte mit Anforderungsbeschreibungen einer Problemsituation im Kontext zu analysieren und aus diesen relevante Informationen herauszufiltern. Dies stellt eine fachübergreifende Kompetenz dar und für die Schülerinnen und Schüler auch außerhalb des Faches Informatik ein Mehrwert.

Um unseren Schülerinnen und Schülern im Laufe der Jahrgänge immer weiter zu ermöglichen, diese Fähigkeit zu erreichen hat sich die Fachgruppe Informatik auf die folgenden Punkte geeinigt.

1.	Für jedes Thema und jeden Jahrgang ein Lexikon mit Fachbegriffen anfertigen. Diese sollten von den Lernenden ausgefüllt werden. Hier sollten auch Wortkompositionen (z. B. Zeichenkette, Sortieralgorithmus, Datenstruktur,...) erklärt, Verben zu Nomen und verschiedene Konjugationen und Deklinationen integriert werden.
2.	Prüfen, ob geschachtelte Sätze aufgeteilt werden können.
3.	Prüfen, ob Sätze, die sich direkt auf andere beziehen, geändert werden können.
4.	Passiv- in Aktivkonstruktionen ändern (z. B. "Die Reihung wird von vorne nach hinten durchlaufen." zu "Man durchläuft die Reihung von vorne nach hinten.").
5.	Rhetorische Fragen zum "Mitdenken" einbauen (z. B. "Wie kannst du jetzt zählen, wie viele Elemente der Stack hat?").
6.	Abbildungen sinnvoll vor oder nach Textabschnitten und weniger daneben stellen. Das erhält die Textordnung.
7.	Unterschiedliche Inhalte sollten eine eigene Überschrift erhalten und so deutlich von anderen getrennt sein. Also gegebenenfalls mehr Abschnitte. Das verbessert die Textgliederung (z. B. Nicht einen einzigen Abschnitt zu DEA, sondern einteilen in z. B. "Einstieg in endliche Automaten", "DEA", "Implementierung eines DEA",...).
8.	Wichtiges kennzeichnen (Farben nutzen, Schriftart, Schriftgröße, Fettdruck,...)

Die Unterstützung wird in den früheren Jahrgängen größer sein und mit steigendem Alter reduziert.

Das Fachwörterbuch wird von den Schülerinnen und Schülern eigenständig fortgeführt und regelmäßig besprochen. Es ist sinnvoll, die Begriffe irgendwann in Kategorien einzuordnen, damit die Schülerinnen und Schüler im Laufe der Schuljahre schneller Begriffe finden können.

Beispiel für eine überarbeitete Fassung eines Unterrichtsinhalts

Das folgende Beispiel ist ein Informationstext aus dem Skript "Einführung in das algorithmische Problemlösen" des Jahrgangs 9.

Die orange dargestellten Wörter sind der Fachsprache zuzuordnen. Hier ist ebenfalls relevant, dass Ausdrücke existieren, die fachsprachlich eine andere Bedeutung haben, als es Lernenden alltags-sprachlich bekannt ist.

Die grau unterlegten Passagen sind nicht nach den Kriterien auf der vorherigen Seite überarbeitet worden.

1. Ursprüngliche Variante

Modellvorstellung einer Variablen

An unserem Eingangsbeispiel wird deutlich, dass sich der Calliope häufig etwas "merken" muss. Er benötigt eine Art Gedächtnis.

Schaut man sich alle Blöcke an, wird man keinen finden, der automatisch irgendeinen Wert hoch- oder runterzählt. Hierfür gibt es beim **Programmieren Variablen**.

Beim Beispiel mit den Eindringlingen erkennt man, dass Variablen für **Algorithmen** sehr wichtig sind. Sie werden für die folgenden Dinge verwendet:

- **Speichern** von **Informationen**, die der Algorithmus später noch benötigt.
- Berechnungen durchzuführen.

Variablen ermöglichen einer **Maschine** also eine Art Gedächtnis zu besitzen.

Sehr vereinfacht kannst du dir vorstellen, dass der Speicher eines **Computersystems** wie ein Schrank mit verschiedenen Schubladen ist (siehe Abbildung rechts).



Variablen in **OpenRoberta** legst du so an:

2. Auf das + Klicken

1.

Name der Variable

Art (Datentyp) der Variable

Startwert

Variable anzahl : Zahl

Zahl
logischer Wert
Zeichenkette
Farbe
Bild
Liste Zahl
Liste logischer Wert
Liste Zeichenkette
Liste Farbe
Liste Bild

Die Variable kannst du dann mit diesen beiden Blöcken verändern oder nutzen:

Schreibe anzahl

anzahl

Wert der Variable anzahl ändern

Variable anzahl nutzen

Wenn die Variable **anzahl** erstellt mit **- Variable anzahl : Zahl** **deklariert** wurde, sucht der Computer eine noch freie Schublade und schreibt sowohl den **Datentyp**, wie auch den Namen der Variablen auf diese Schublade (Abbildung unten links).

Jeder mit **- Variable anzahl : Zahl** **0** deklarierten Variablen wird bei Open Roberta Lab direkt ein **Startwert** zugewiesen. Diesen Vorgang nennt man **initialisieren**. Der Computer öffnet nun die entsprechende Schublade und legt den Startwert, bei uns 0, hinein (Abbildung unten rechts).



Deklarieren der Variable `anzahl` vom Datentyp `zahl`



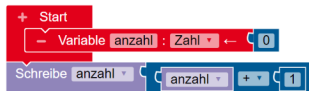
Initialisieren der Variable `anzahl` mit dem Wert 0

Anschließend kannst du mit `Schreibe anzahl ← 2` den Wert der Variablen `anzahl` auf 2 setzen. Der Computer sucht nun die Schublade, auf der `laenge` steht, öffnet sie und legt die Zahl 2 hinein. Anschließend schließt der die Schublade wieder.

Der Slot bei `Schreibe anzahl ← " " "` ist blau. Man kann also tatsächlich nur Blöcke, die Zahlen sind, aus dem ebenfalls blauen Bereich Mathematik anhängen und nicht den grünen für ein **Zeichen** oder eine **Zeichenkette**.



Wenn du den Wert von `anzahl` ändern möchtest, kannst du das mit dem `setze-Block` direkt, wie oben, tun oder mit dem Block:



Bei "Schreibe `anzahl` auf `anzahl` + 1" sucht sich der Computer die Schublade mit dem Namen `anzahl` heraus, öffnet sie und kopiert sich die darin liegende Zahl heraus.

Anschließend wird der neue Wert von `anzahl` wieder in die Schublade gelegt und damit der alte **überschrieben**. (siehe die drei Abbildungen am Ende dieser Seite)

Als Name einer Variablen ist im Prinzip alles erlaubt, z.B.: `xyz`, `test`, `i`, `x`. In größeren Programmen verliert man da leicht den Überblick. Wähle deshalb am besten immer einen möglichst aussagekräftigen Namen, der einen Hinweis darauf gibt, was in der Variablen gespeichert ist.



2. Erste Überarbeitung

Die Sätze wurden nach den Kriterien in der Tabelle überarbeitet. Außerdem wurden die Abbildungen lesefreundlicher gesetzt.

Modellvorstellung einer Variablen

An unserem Eingangsbeispiel wird deutlich, dass sich der Calliope häufig etwas "merken" muss. Er benötigt eine Art Gedächtnis.

Es gibt keinen Block, der automatisch irgendeinen Wert hoch- oder runterzählt. Hierfür gibt es beim Programmieren **Variablen**.

Schauen wir uns das Beispiel mit den Eindringlingen an. Dort ist eine Variable sehr wichtig. Mit einer kann der Algorithmus die Anzahl der Eindringlinge zählen. Variablen werden allgemein für die folgenden Dinge verwendet:

- Speichern von Informationen, die der Algorithmus später noch benötigt.
- Berechnungen durchzuführen.

Variablen ermöglichen einer Maschine also eine Art Gedächtnis zu besitzen.

Sehr vereinfacht kannst du dir vorstellen, dass der Speicher eines Computersystems wie ein Schrank mit verschiedenen Schubladen ist (siehe Abbildung rechts).



Variablen in OpenRoberta legst du so an:

Die Variable kannst du dann mit diesen beiden Blöcken verändern oder nutzen:

Bei der Erstellung der Variable `anzahl` mit dem Block `- Variable anzahl : Zahl` geht der Computer in den folgenden Schritten vor:

Diese drei Schritte werden **Deklarieren** genannt.

1. Der Computer sucht eine noch unbeschriftete Schublade.
2. Der Computer schreibt den **Datentyp**, zum Beispiel **Ganzzahl**, auf ein Schild der Schublade.
3. Der Computer schreibt den Namen der Variablen auf ein weiteres Schild dieser Schublade.

Jeder mit  deklarierten Variablen wird bei Open Roberta Lab direkt ein Startwert zugewiesen. Diesen Vorgang nennt man **Initialisieren**.

Initialisieren geschieht in den folgenden Schritten:

1. Der Computer öffnet die Schublade mit dem angegebenen Namen.
2. Der Computer überprüft, ob der Wert, den er hineinlegen soll, dem Schild für den Datentyp auf der Schublade entspricht.
3. Der Computer legt den Startwert, bei uns 0, hinein.

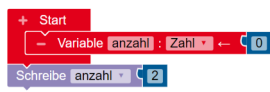
Beide Vorgänge sind in den Abbildungen unten dargestellt.



Deklarieren der Variable `anzahl` vom Datentyp `zahl`



Initialisieren der Variable `anzahl` mit dem Wert 0

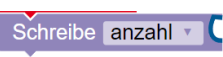
Anschließend kannst du mit  den Wert der Variablen `anzahl` auf 2 setzen. Das geschieht in den folgenden Schritten:

1. Der Computer sucht die Schublade, auf der `laenge` steht.
2. Der Computer öffnet die Schublade.
3. Der Computer legt die Zahl 2 hinein.

Hinweis: Die Zahl, die vorher in der Schublade war, verschwindet bei diesem Vorgang.

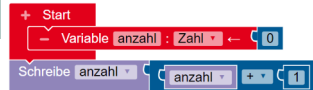


Beispiel für eine Wertzuweisung der Variablen `anzahl`

Achtung: Der "Slot" rechts beim Block  ist blau.

Merke dir: Man kann also tatsächlich nur Blöcke, die Zahlen sind, aus dem ebenfalls blauen Bereich Mathematik anhängen.
Der Grund ist, weil du die Variable `anzahl` mit dem Datentyp `Zahl` deklariert hast.
Wenn du eine Variable als ein **Zeichen** oder eine **Zeichenkette** deklariert hättest, könntest du nur den grünen Block verwenden.

Es gibt noch eine weitere Möglichkeit, wenn du den Wert von `anzahl` ändern möchtest.
Nämlich mit dem Block:



Bei "Schreibe `anzahl` auf `anzahl` + 1" geht der Computer in den folgenden Schritten vor:

1. Der Computer sucht die Schublade mit dem Namen `anzahl` und öffnet sie.
2. Der Computer kopiert sich die darin liegende Zahl heraus.
3. Der Computer berechnet den neuen Wert der Variable `anzahl`.
4. Der Computer legt den neuen Wert von `anzahl` in die Schublade.
5. Der vorherige Wert in der Schublade wird dabei überschrieben.

Die drei Abbildungen weiter unten stellen den Vorgang dar.

Als Name einer Variablen ist im Prinzip alles erlaubt, z.B.: `xyz`, `test`, `i`, `x`. In größeren Programmen verliert man da leicht den Überblick. Wähle deshalb am besten immer einen möglichst aussagekräftigen Namen, der einen Hinweis darauf gibt, was in der Variablen gespeichert ist.



3. Zweite Überarbeitung

In den obigen Varianten werden eine vereinfachte Modellvorstellung von Variablen und die Anwendung in der Programmierumgebung parallel eingeführt. Dies erscheint deutlich schwieriger, als beides zu trennen. Weiterhin wurde der Textumfang reduziert und die verschiedenen Aufzählungen entfernt.

Modellvorstellung einer Variablen

An unserem Eingangsbeispiel wird deutlich, dass sich der Calliope häufig etwas "merken" muss. Er benötigt eine Art Gedächtnis.

Schaut man sich alle Blöcke an, wird man keinen finden, der automatisch irgendeinen Wert hoch- oder runterzählt. Hierfür gibt es beim Programmieren **Variablen**.

Beim Beispiel mit den Eindringlingen erkennt man, dass Variablen für Algorithmen sehr wichtig sind. Sie werden für die folgenden Dinge verwendet:

- Speichern von Informationen, die der Algorithmus später noch benötigt.
- Berechnungen durchzuführen.

Variablen ermöglichen einer Maschine also eine Art Gedächtnis zu besitzen.

Sehr vereinfacht kannst du dir vorstellen, dass der Speicher eines Computersystems wie ein Schrank mit verschiedenen Schubladen ist.



Wenn man eine Variable erstellt, fachsprachlich nennt man das **deklariert**, muss dem Computer immer ein **Datentyp** und der Name der Variable gegeben werden.

Ein Datentyp sagt dem Computer, welche Art von Daten unter dem Namen der Variablen gespeichert werden darf. Für unser Beispiel ist die Variable eine **Zahl**, da wir etwas damit zählen lassen müssen. Und unsere Variable soll den Namen `anzahl` bekommen.

Jetzt sucht der Computer eine noch freie Schublade und schreibt sowohl den Datentyp, wie auch den Namen der Variablen auf diese Schublade.



Deklarieren der Variable `anzahl` vom Datentyp `zahl`

Jeder in OpenRoberta deklarierten Variablen wird ein **Startwert** zugewiesen. Diesen Vorgang nennt man **initialisieren**. Der Computer öffnet nun die entsprechende Schublade und legt den Startwert, bei uns 0, hinein.



Initialisieren der Variable `anzahl` mit dem Wert 0

Nun sagen wir dem Computer, dass er den Wert der Variablen `anzahl` auf 2 setzen soll. Dazu sucht der Computer die Schublade auf der `anzahl` steht, öffnet diese und legt die Zahl 2 hinein. Anschließend schließt der die Schublade wieder.



Setzen der Variable `anzahl` auf den Wert 2

Achtung: Das Beispiel mit dem Setzen der Variablen `anzahl` auf den Wert 2 funktioniert nur, weil "2" eine Zahl ist.

Würde man versuchen der Variablen `anzahl` ein einzelnes Zeichen, z. B. 'a', oder eine Zeichenkette, z. B. "Hallo Computer!", als neuen Wert zuweisen, funktioniert das nicht. Der Computer sucht nämlich nicht nur die Schublade mit dem Namen `anzahl`, sondern er überprüft auch, ob der Datentyp auf der Schublade zum übergebenen Wert passt.

Der Computer soll eine Berechnung, z. B. die Erhöhung des aktuellen Wertes der Variablen `anzahl` um 1, durchführen. Diese Anweisung kann man sich so vorstellen:

Anweisung:	<code>anzahl</code>	$\leftarrow$	<code>anzahl</code>	+	1
Interpretation:	Der neue Wert von <code>anzahl</code> ; soll sein; der aktuelle Wert von <code>anzahl</code> ; erhöht um 1.				

Mit dieser Anweisung sucht sich der Computer die Schublade mit dem Namen `anzahl` heraus, öffnet sie und kopiert sich die darin liegende Zahl heraus.

Anschließend wird der neue Wert von `anzahl` wieder in die Schublade gelegt und damit der alte überschrieben. (siehe die drei Abbildungen unten)



Berechnung eines neuen Wertes für die Variable `anzahl`: $\text{anzahl} \leftarrow \text{anzahl} + 1$

Als Name einer Variablen ist im Prinzip alles erlaubt, z.B.: `xyz`, `test`, `i`, `x`. In größeren Programmen verliert man da leicht den Überblick. Wähle deshalb am besten immer einen möglichst aussagekräftigen Namen, der einen Hinweis darauf gibt, was in der Variablen gespeichert ist.

Variablen in Open Roberta

Deklarieren und Initialisieren:

The screenshot shows the Open Roberta workspace. On the left, the 'Variablen' category is selected, indicated by a red box and the number '1.'. A red arrow points to the '+' icon in the 'Start' block, labeled '2. Auf das + Klicken'. The 'Start' block contains a variable declaration: 'Variable anzahl : Zahl'. A red box highlights the 'Start' block, labeled 'Name der Variable'. A red arrow points to the 'Zahl' dropdown menu, labeled 'Art (Datentyp) der Variable'. A red arrow points to the '0' value, labeled 'Startwert'. A dropdown menu is open, showing options: 'Zahl', 'logischer Wert', 'Zeichenkette', 'Farbe', 'Bild', 'Liste Zahl', 'Liste logischer Wert', 'Liste Zeichenkette', 'Liste Farbe', and 'Liste Bild'.

Die Variable kannst du dann mit diesen beiden Blöcken verändern oder nutzen:

Two blocks are shown: 'Schreibe anzahl' (labeled 'Wert der Variable anzahl ändern') and 'anzahl' (labeled 'Variable anzahl nutzen').

Anschließend kannst du mit den Wert der Variablen `anzahl` auf 2 setzen.

Wenn du den Wert von `anzahl` durch eine Berechnung ändern möchtest, kannst du das mit diesem Block tun:

Der Slot bei ist blau. Man kann also tatsächlich nur Blöcke, die Zahlen sind, aus dem ebenfalls blauen Bereich Mathematik anhängen (siehe Hinweis im orangenen Kasten beim vorherigen Abschnitt zur Modellvorstellung von Variablen).

(Zeitplanung: ca. 15 Doppelstunden)

[Zum Anfang zurück](#)

Grundlagen der Datenverarbeitung - Codierung von Daten - Dualzahlen, ASCII, EAN-8, n-bit-Repetitionscode und verlustbehaftete Verfahren. - Diverse Zusatzcodierungen	5W	1.-2. (OM 1.4-6.2) (BO) Einstieg in notwendige Absprachen bei der Kommunikation (Flaggenszenario atGA) mit Ergebnis: Kommunikationsprotokolle und Motivation einer einheitlichen Codierung von Zeichen → Datenspeicherung und -verarbeitung technisch (Motivation zu Dualzahlen) → Das Binärsystem (Umwandlung in beide Richtungen, optional: Addieren von Dualzahlen) → Der ASCII-Code → atGA: n-bit Repetitionscode und ein verlustbehaftetes Kompressionsverfahren → Zusatzaspekte: Farbbilder speichern, RGB, Braille, EAN-8-Code, Nachrichtenprotokolle zu ASCII mit dem Calliope. P1.2-5, P3, P4, I1.1, I1.3, I1.4, I4.1-3	Lernfeld „Computerkompetenz“: - beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip). - beschreiben die Hardwarekomponenten eines Computers und ihre Funktion. - benennen verschiedene Speichermedien und Speicherorten und erläutern die Unterschiede. - erläutern die Funktionsweise von verschiedenen Arten von Ein- und Ausgabege-räten. Lernfeld „Daten und ihre Spuren“: - erläutern die Notwendigkeit, Daten in geeigneter Form zu codieren, um sie mit dem Computer verarbeiten zu können. - nennen Beispiele für die Codierung von Daten wie Morsecode, ASCII. - codieren und decodieren Daten mithilfe eines vorgegebenen Verfahrens. - erläutern Codierungsverfahren für Bilddaten. - erläutern einfache Verfahren der Datenkompression, z. B. Lauflängenkodierung. - erläutern die Vor- und Nachteile verlustfreier und verlustbehafteter Kompression von Daten. - erläutern, wie Farben mithilfe des RGB- und des CMYK-Modells dargestellt werden. - entwerfen eine Codierung zur Übertragung von Daten über einen Kommunikati-onskanal. - entwerfen ein Protokoll zur Übertragung von Daten über einen Kommunikations-kanal.
--	-----------	---	---

Inhalte Jahrgang 10

(Zeitplanung: ca. 15 Doppelstunden)

Klasse 10			
Inhalte/Themen	Zeit	Hinweise zur Umsetzung	Bezug zum KC
Aufbau des Internets - Der Aufbau des Internets - zentrale Komponenten - Client, Server, Router, DNS - Dienste im Internet - Aufbau eines Netzwerkes mit den zentralen Komponenten	3W	Material, Methodik und Differenzierung: siehe Skripte zum Jg. 10 im Gruppenordner Informatik 1. bis 3. (OM 5.2.1-3, 5.4.3, 5.5.3, 6.3) (BO) Einstieg: Zu einem Szenario sollen die Lernenden in Gruppen ihre Vorstellungen der Wege im Internet skizzieren → Präsentation der Ergebnisse → Welche Komponenten und Dienste sind tatsächlich im Internet vorhanden → Beschreibung der Komponenten und Dienste → Erneutes Erstellen der Wege im Internet zu den vorherigen Szenarien mit fertigen Komponentenkarten → Auswertung der Ergebnisse → Lokale Netzwerke zuhause und in der Schule P1.4, P3.1, P3.3, P4.3, P4.4, I3.3, I4.3, I4.4	Lernfeld „Daten und ihre Spuren“: - beschreiben und begründen den dezentralen Aufbau des Internets. - nennen die zentralen Komponenten des Internets, z.B. Client, Server, Router, DNS, und erläutern ihre Funktion. - nennen Maßnahmen wie z. B. Schutz durch Passwörter oder Verschlüsselung, um sicher in Netzwerken zu kommunizieren und Daten vor Fremdzugriff zu schützen. - beschreiben und kategorisieren Nutzungsmöglichkeiten des Internets. - nennen mögliche Formen des Datenmissbrauchs.
Algorithmisches Problemlösen - Variablen mit dem Calliope - Wiederholung zum algorithmischen Problemlösen mit einer grafischen Programmiersprache (Scratch) - Ein eigenes Projekt algorithmisch umsetzen - Eigene Blöcke (Operationen)	3W 3W 2W	1. (OM 5.1+2-8.3) (BO) Einstieg: Notwendigkeit von Variablen im Kontext → Umsetzung in Open Roberta Lab → Verschiedene Übungen (Lichtschalter, Alarmanlage, Variablen zum Zählen und Rechnen, Nachrichtenaustausch mit Variablen, eigene Ideen modellieren und umsetzen) 2 und 3. (OM 5.1, 5.4, 6.3) (BO) Kleine Aufgaben zur Auffrischung → Konkrete Projektideen im Kontext werden vorgestellt, aus denen die Lernenden individuell auswählen können. Es kann auch ein eigenes Projekt realisiert werden. 4. (OM 5.1, 5.4, 6.3) (BO) Einstieg in eigene Blöcke (Motivation: Übersichtlichkeit, Änderungen, Fehler beheben) → Eigene Blöcke zum Zeichnen eines eigenen Bildes erstellen	Lernfeld „Algorithmisches Problemlösen“: - strukturieren Handlungsabläufe in logische Teileinheiten. - benennen typische Bestandteile von algorithmischen Abläufen wie z. B. eindeutige Handlungsanweisungen, Wiederholungen, Verzweigungen. - beschreiben einen gegebenen Algorithmus in eigenen Worten. - überprüfen, ob eine Implementierung die Problemstellung löst.

		Software: Scratch und Open Roberta Hardware: PC P1.1-2, P2, P3.1, P4.1, I2.1-2	- entwerfen einen Algorithmus unter zielgerichteter Verwendung der elementaren Kontrollstrukturen. - formulieren Bedingungen mithilfe der logischen Verknüpfungen „Und“, „Oder“ und „Nicht“. - verwenden Variablen und Wertzuweisungen in einfachen Algorithmen. - stellen einen Algorithmus grafisch dar. - führen einen gegebenen Algorithmus aus. - stellen einen gegebenen Algorithmus in einem Struktogramm dar. - interpretieren ein vorgegebenes Struktogramm. - zerlegen einen komplexeren Algorithmus in mehrere Operationen, um z. B. Teillösungen wiederzuverwenden.
Verwaltung von Daten 1. Daten in Tabellen verwalten 2. Grundlegende Begriffe zu Datenbanken, Redundanzen, Anomalien und Inkonsistenzen 3. Erste Grundlagen zu SQL Abfragen	4W	1. bis 3. (OM 5.1, 5.4, 6.3) (BO) Einstieg: Datenverwaltung im Kontext Clubs → Grundlagen zum Datenbankentwurf mit mehreren Tabellen → Begriffe: Datenbank, Datenbankmanagementsystem, Tabelle, Schema, Attribut, Datensatz, Attributwert, Schlüsselattribut, Primärschlüssel → Einstieg: Redundanzen, Anomalien (Einfüge-, Änderungs- und Löschanomalie), Inkonsistenzen an konkreten Beispielen → Projektion und Selektion mit SQL → Einfache Aggregatfunktionen → Übungen mit SQL Tutorial L1+2 → Abfragen über mehrere Tabellen (Join) → Übungen mit SQL Tutorial L3 Software: SQL-Datenbank online (SQL-Tutorial) Hardware: PC P1, P3.1-2, P4, P5.1-2, I1.3	Lernfeld „Daten und ihre Spuren“: - nennen Kriterien zur Strukturierung von Daten. - erstellen Tabellen zur Strukturierung von Daten mithilfe einer Tabellenkalkulation oder eines Datenbanksystems. - unterscheiden Datensatz, Attribut, Schlüsselattribut, Tabelle und Datenbank. - erläutern die Notwendigkeit, Datensätze eindeutig identifizieren zu können (Primärschlüssel). - nutzen Schlüsselattribute zur Verknüpfung von Daten. - gewinnen Informationen aus den Daten einer Tabellenkalkulation (oder Datenbank) z. B. durch Filtern und Sortieren. - unterscheiden zwischen Informationen und ihrer Repräsentation durch Daten. - stellen Daten in Form von Diagrammen grafisch dar. - formulieren einfache Suchanfragen an Datenbanken. - formulieren einfache Suchanfragen an Datenbanken über mehrere Tabellen. - formulieren Suchanfragen an Datenbanken auch durch Verwendung von Aggregatfunktionen. - erläutern die rechtlichen Rahmenbedingungen für den Umgang mit ihren persönlichen Daten wie z.B. informationelle Selbstbestimmung, Allgemeine Geschäftsbedingungen (AGB) und Datenschutz

Inhalte der Einführungsphase

Die Klassen der E-Phase sind eine Woche im Praktikum. Die zusätzliche Unterrichtszeit, die nach KC ca. 25 % der Gesamtzeit ausmacht, kann eigenständig mit Inhalten gefüllt werden. Zeitplanung: 38 Wochen - 10% → ca. 33 Doppelstunden

Klasse 11			
Inhalte/Themen	Zeit	Hinweise zur Umsetzung	Bezug zum KC
Informationen und Daten Codierungen 1. Ziele der Codierungstheorie 2. Verschiedene Codierungen - Lauflänge, Prüzfiffer, RGB 3. Einführung in die Programmierumgebung Processing (Java) 4. Kontrollstrukturen und Variablen (int, char, String, boolean) in Processing (Java)	8W	1. und 2. (OM 1.4-5.2, 5.3+5.2+3) (BO) Ziele der CT durch Vergleich von ASCII, Morse und einem 3-bit-Gencode → Lernstationen zu verschiedenen neuen Codierungen (Lauflängencodierung bei Graubildern, Prüzfiffer, Codebäume, Lauflängencodierung bei Farbbildern, Entropiecodierungen, Vertiefungen RGB, Farbbildcodierung mit Scratch) → Übergang zu Processing: Wiederholung Kontrollstrukturen und Variablen Aufgaben zu Variablen und Datentypen (String, char, int, boolean, inklusive Deklarieren / Initialisieren, Vergleichen, Ausgabe) in Verbindung mit Kontrollstrukturen in Processing → Algorithmische Umsetzung einer Codierung (Repetitionscode) Software: Processing, LibreOffice Hardware: PC PK1.1, PK1.2, PK2.1, PK2.2, PK3.2 IK1.2, IK2.1, IK2.2	Lernfeld „Informationen und Daten“: - beschreiben grundlegende Codierungen von Daten, u. a. Dualzahlen, ASCII, RGB-Modell. Lernfeld „Algorithmen und Datenstrukturen“: - entwerfen und implementieren Algorithmen unter zielgerichteter Verwendung der elementaren Kontrollstrukturen. - verwenden Variablen und Wertzuweisungen in Algorithmen. - entwerfen und implementieren Algorithmen unter Verwendung elementarer Zeichenkettenoperationen.
Algorithmen und Datenstrukturen 1. Einlesen und Verarbeiten von Eingaben (Statischer und Aktiver Modus) 2. Interaktion Maus / GUI 3. Projekt: Vokabeltrainer 4. Algorithmen standardisiert darstellen - Struktogramme 5. Tracetabellen - Verfolgung von Variablenbelegungen mit Hilfe von Tracetabellen.	7W 2W 2W	Material, Methodik und Differenzierung: siehe Skripte zum Jg. 11 im Gruppenordner Informatik 1. bis 5. (OM 5.1-6.3) (BO) Anforderungsbeschreibung und Rahmenbedingungen zum Projekt: Vokabeltrainer → Einlesen von Usereingaben (Statischer und Aktiver Modus) → Zufallszahlen → Projekt: Vokabeltrainer implementieren 6. (OM 5.1, 5.4, 6.3) (BO) Einführung in Nassi-Shneidermann Struktogramme (Motivation: Austausch zu Programmen in verschiedenen Programmiersprachen) → Entwerfen eigener Algorithmen in Form eines Struktogramms im Szenario "Mäusefangen" → Struktogramme analysieren und erweitern bzw. korrigieren 7. (OM 5.1-6.3) (BO) Einstieg und Motivation: Code-Mainia Kartenspiel → Einführung in Tracetabellen → Übungen zu Tracetabellen, inklusive Algorithmen in Struktogrammform Software: Processing, LibreOffice Hardware: PC PK1.1, PK1.2, PK2.1, PK2.2, PK3.2 IK1.2, IK2.1, IK2.2	Lernfeld „Algorithmen und Datenstrukturen“: - benennen Anweisung, Sequenz, Schleife und Verzweigung als Grundbausteine eines Algorithmus. - entwerfen und implementieren Algorithmen unter zielgerichteter Verwendung der elementaren Kontrollstrukturen. - stellen Algorithmen in standardisierter Form dar. - erläutern das Prinzip der Speicherung von Werten in Variablen. - stellen die Belegung von Variablen bei der Ausführung eines Algorithmus in Form einer Tracetabelle dar. - verwenden Variablen und Wertzuweisungen in Algorithmen. - entwerfen und implementieren Algorithmen unter Verwendung elementarer Zeichenkettenoperationen. - stellen einen Algorithmus grafisch dar.
Datenaustausch in Netzwerken 1. Komponenten eines Netzwerks als Informatiksystem - Endgeräte, Switch, IP, Router, DNS, Web-Server, E-Mail-server	1W	1. (OM 1.4-5.2, 5.3+5.2+3) (BO) Kommunikation im Internet und Netzwerken (Webadressen, Topographien, Client, IP-Adressen) → Szenario mit Clients, Switch, Router, DNS, Web-Server (Einzelne Aspekte sollen mit den Erkenntnissen der vorherigen Sequenz geprüft und gegebenenfalls korrigiert werden). Software: LibreOffice, WebNetSim (BO) Hardware: PC	Lernfeld „Informationen und Daten“: - beschreiben grundlegende Codierungen von Daten, u. a. Dualzahlen, ASCII, RGB-Modell. - beschreiben zentrale Komponenten eines Informatiksystems und deren Zusammenspiel. - beschreiben und begründen den dezentralen Aufbau des Internets. - nennen die zentralen Komponenten des Internets, u. a. Client, Server, Router, DNS und erläutern ihre Funktion.
Datenanalyse und DSGVO 2. Datenanalyse und automatisierte Datenanalyse 3. Chancen und Risiken automatisierter Datenanalyse 4. Rechtliche Rahmenbedingungen im Umgang mit Daten - informationelle Selbstbestimmung / DSGVO	6W	2. bis 4. (OM 2.2+3.2+3, 4.1-4.1-3, 6.3+4.1, 6.1+4+5.2, 6.4+5.3) (BO) Einstieg: Artikel mit Diskussion zur Supermarktkette Target USA (OM 4.1.1-3, 4.2.2-3, 4.3.1+3, 6.3.1-2, 6.4.1, 6.1.3) → Korrelation und Kausalität → Der Prozess der Datenanalyse → Klassifikationsbäume mit manueller Datenanalyse (OM 3.1+3.1, 3.2.3, 5.2.2, 5.4-6.3) Projekt ca. 2 Wochen: Automatisierte Datenanalyse mit Orange → Interpretation der Ergebnisse → Analyse: Chancen und Risiken automatisierter Datenanalyse atGA (OM 1.1-6.1, 1.2-5.3, 2.2-4.2, 3.1-4.3, 4.1.1-3, 4.2.2-3, 4.3.1+3, 5.2-6.3, 6.4-5.3) GA zum Gesetz „Recht auf informationelle Selbstbestimmung“ mit Gruppenpräsentationen zu	- beschreiben die Kommunikationswege im Internet. - beschreiben Aspekte zur Sicherheit der Kommunikation im Internet. - erläutern die rechtlichen Rahmenbedingungen für den Umgang mit ihren persönlichen Daten, wie z. B. informationelle Selbstbestimmung und Datenschutzrichtlinien. - beschreiben das Prinzip der Transposition und der Substitution zur Verschlüsselung von Daten.

Kryptologie 5. Transpositions- und Substitutionsverfahren 6. Implementationen verschiedener Verschlüsselungsverfahren mit Processing - Verwendung und elementare Operationen von Zeichenketten 7. Sicherheit bei einfachen Verschlüsselungen - Häufigkeitsanalyse, inklusive algorithmischer Umsetzung in Processing 8. Sicherheit der Kommunikation im Internet und sichere Passwörter	7W	einzelnen relevanten Aspekten → atGA zu den Grundsätzen des Datenschutzes anhand der DSGVO Material, Methodik und Differenzierung: siehe Skript zum Jg. 11 im Gruppenordner Informatik Software: LibreOffice, Orange Hardware: PC PK3.3 IK1.1, IK3.1, IK3.3, IK4.2 5. bis 8. (OM 2.1.2, 2.2-3.3, 3.1.3, 3.2.2, 2.4.1, 6.3-4.2, 6.5.3) (BO) Verschiedene Verschlüsselungen im Stationenlernen → Klassifizierung nach Transpositions- und Substitutionsverfahren → Implementationen verschiedener Verfahren (Hier wird mit Zeichenketten gearbeitet) → Kryptoanalyse bei den bekannten Verfahren (u. a. Häufigkeitsanalyse), inklusive Implementationen → Escape-Room Krypto → (optional) Sicherheit durch Passwörter (Material von inf-schule.de) Software: Processing, cryptool oder gc.de Hardware: PC PK2.2, PK3	- implementieren monoalphabetische Verfahren, u. a. Caesar-Verfahren. - erläutern das Prinzip der Häufigkeitsanalyse. - beurteilen die Sicherheit einfacher Verschlüsselungsverfahren. Lernfeld „Algorithmen und Datenstrukturen“: - entwerfen und implementieren Algorithmen unter zielgerichteter Verwendung der elementaren Kontrollstrukturen. - verwenden Variablen und Wertzuweisungen in Algorithmen. - entwerfen und implementieren Algorithmen unter Verwendung elementarer Zeichenkettenoperationen. - nennen Maßnahmen wie z. B. Schutz durch Passwörter oder Verschlüsselung, um sicher in Netzwerken zu kommunizieren und Daten vor Fremzugriff zu schützen. Lernfeld „Informationen und Daten“: - beschreiben und kategorisieren Nutzungsmöglichkeiten des Internets. - nennen mögliche Formen des Datenmissbrauchs. - diskutieren die Chancen und Risiken der automatisierten Datenanalyse.
---	---------------	---	---

		<u>5. (OM 5.1-6.3)</u> Einstieg ADT Baum (atGA mit 1. Binäre Suche: Konsolenquartett sortieren, Anordnen der Karten in Baumstruktur und Nim-Spiel mit Streichhölzern) (BO) → Fachbegriffe zum ADT Baum (Unterschied zu linearen ADT, Knoten, Kante, Wurzel, Blatt, Tiefe, Ebene) → Übungen zum ADT Baum (z. B. Grad und Tiefe in vorgegebenen Bäumen bestimmen, gegebene Strukturen als Baum klassifizieren, Wortergänzung...) → Einstieg in Binärbäume (z. B. an einem Stammbaum) → Saubere rekursive Definition des ADT BinaryTree → Klasse das ADT Binärbaum / BinaryTree sichten → verschiedene Konstruktoren des ADT BinaryTree → Erstellen eines Objektes vom Typ Binary-Tree (z. B. einen konkreten Stammbaum) → Ausgabe der Inhalte eines Binärbaums (rekursiv) → Traversierungen (in-, pre- und postorder inklusive Implementierung in die Klasse BinaryTree) (BO) → Übungen zum Erstellen von Binärbäumen, Traversierungen, aus Traversierungen Binärbäume rekonstruieren → Suchen (inklusive Implementation), Einfügen (inklusive Implementation) und Entfernen in Binären Suchbäumen 2W Software: Java-Editor, y-ED und LibreOffice Writer (BO) <u>6. (OM 5.1-6.3)</u> Mergesort → Divide and Conquer Prinzip → Türme von Hanoi → Quicksort → Erweiterung der Laufzeituntersuchungstabelle der Sortieralgorithmen Software: Java-Editor und LibreOffice Writer (BO)	
Codierung / Kryptologie 1. Verlustfreie Datenkompression bei Texten und Grafiken: Huffman-Code, Lauflängencodierung, kombinierte Bildkompression 2. Fehlererkennende Codes (Repetitionscode, (7,4) Hamming-Code) 3. Einsatzgebiete für Verschlüsselungen (z.B. E-Mail), historische Entwicklung 4. Kryptologische Verfahren - Caesar und Vigenère, symmetrische und asymmetrische Verfahren und Schlüsseltausch, digitale Signaturen, Zertifikate), DES, Blockchiffre	5W 4W 2W 7W 6W	<u>1.</u> Wiederholung: Ziele der CT und RGB → Programmierprojekt zur Lauflängencodierung → Einstieg in Kompressionsverfahren: Huffmanncodierung von Texten (OM 1.4.2, 5.1-5.3) (BO) → Übungen zur Textkompression mit Huffman → (atPA) Farbbildcodierung mit Lauflänge und Huffman (BO) → Eigenschaften und Vergleich der beiden Codierungen (Optimale bzw. schlechte Bilder für die jeweilige Codierung, Optimale Codierung) (OM 1.4.2, 5.1-5.3) (BO) → Kombinierte Bildkompression (BO) → Projekt: Algorithmische Vorgehensweise bei der Erstellung des Huffmanbaums (Implementierung mit DynArrays und 2d-Arrays, Zeichen zählen, Sortierung, Einfügen in nach Wurzeln von BinTrees, in ein DynArray sortieren, zusammensetzen und wieder Einsortieren, Tiefe in BinTree ergänzen) → Algorithmische Umsetzung verlustbehafteter Verfahren Software: Java-Editor, LibreOffice Writer/Calc und yEd (BO) <u>2.</u> Wiederholung Fehlererkennung und Fehlerkorrektur (mit z. B. einem Repetitionscode) → Implementierung einer Repetitions-codierung (inklusive Übertragungsfehlererkennung / Korrektur) → arbeitsteilige Gruppenarbeit mit digitaler Präsentationswand: Fehlererkennung im Alltag (z. B. Eurobanknoten, Kreditkarten, QR, ISBN, EAN-Barcode (Recherchieren und Präsentation) (OM 1.1-3+5-6.3, 5.1-6.3) (BO) → (7,4) Hamming-Code → Implementierung des (7,4) Hamming-Codes (OM 1.4.2, 5.1-5.3) → Abituraufgaben zur Codierungstheorie (gA/eA) Software: Java-Editor, LibreOffice Writer/Calc, Präsentationssoftware und yEd (BO) PK1.1, PK2.1, PK2.2, PK3.3, IK1.1, IK2.3 , IK3.4 <u>3. und 4.</u> Wiederholung klassischer Verfahren (monoalphabetische Substitution (Caesar), Transpositionsverfahren, Häufigkeitsanalyse) → Präsentation zu Einsatzgebieten für Verschlüsselungen (Angriffstypen bei Kommunikation, Gefahren in Netzwerken, Sender-Empfänger, Kanal, Schlüssel, Verschlüsselungen bei Emails) (OM 2.1.2, 2.2-3.3, 3.1.3, 3.2.2, 2.4.1, 6.3-4.2, 6.5.3) (BO) → Vigenère → Kryptoanalyse bei Vigenère (Warum funktioniert eine einfache Häufigkeitsanalyse nicht, Kasiski Test) → Übungen zum Ver- und Entschlüsseln mit Vigenère, Kasiskitest, Projekt: Implementierung zu Vigenère (BO), XOR mit Implementierung → Blockchiffre am Beispiel des DES (vereinfacht mit nur zwei Runden und 8 Bit-Blöcken) → Programmierprojekt: DES → Das Schlüsseltauschproblem → Asymmetrische Verfahren – Präsentation (Grundlegende Aspekte asymmetrischer Verfahren) (BO), digitale Signaturen (Authentifizierung) → digitale Zertifikate (OM 2.1.2, 2.2-3.3, 2.4.1, 4.1.3, 4.2.2, 4.3.3, 5.3.3) → Digitale Unterschrift mit Verschlüsselung und Authentifizierung → Hashfunktionen für digitale Signaturen (inklusive Entwicklung eines Algorithmus) (OM 2.1.2, 2.2-3.3, 2.4.1, 4.1.3, 4.2.2, 4.3.3, 5.3.3) (BO) Software: Java-Editor und LibreOffice Writer (BO) Hardware: PC PK2.1, PK2.2, PK3.3, IK1.1, IK2.3, IK4.1, IK4.3	- beschreiben Möglichkeiten, Daten zu komprimieren, u. a. Lauflängencodierung, Huffman-Codierung. - erläutern Möglichkeiten der Fehlererkennung und der Fehlerkorrektur bei der Datenübertragung, u. a. Paritätsbit, (7,4)-Hamming-Code. - beschreiben das Prinzip der polyalphabetischen Substitution, u. a. am Beispiel des Vigenère-Verfahrens. - beurteilen die Sicherheit eines gegebenen symmetrischen Verschlüsselungsverfahrens. - beschreiben und unterscheiden die Prinzipien der symmetrischen und asymmetrischen Verschlüsselung. - beschreiben Anwendungsbereiche für symmetrische bzw. asymmetrische Verschlüsselungsverfahren. - entwerfen und implementieren ein symmetrisches Verschlüsselungsverfahren. - erläutern die prinzipielle Funktionsweise eines modernen symmetrischen Blockchiffreverfahrens. - erläutern das Prinzip von digitalen Signaturen und Zertifikaten. - entwerfen und implementieren ein Kompressionsverfahren zu einem gegebenen Sachverhalt. - erläutern die Vor- und Nachteile verlustfreier Kompression von Daten. - entwerfen und implementieren ein Protokoll zur Übertragung von Daten über einen Kommunikationskanal - diskutieren die Chancen und Risiken der automatisierten Datenanalyse.

Datenbanken 1. Wiederholung zu Datenbanken 2. Beschreibung der Wirkungsweise grundlegender SQL-Abfragen zur Datenbankauswertung anhand eines konkreten Satzes von Relationen 3. Entwurf und Anwendung einfacher Abfragen und Verbundabfragen in SQL 4. Entwurf und Anwendung geschachtelter SQL- Abfragen 5. Entwicklung eines ER-Modells für ein vorgegebenes System 6. Analyse eines vorgegebenen ER-Modells bezüglich eines Anwendungsfalls 7. Erweiterung eines vorgegebenen ER-Modells 8. Umsetzung eines ER-Modells in ein relationales Datenbankschema 9. Entwicklung eines relationalen Datenbankschemas unter Berücksichtigung der ersten, zweiten und dritten Normalform	2W 4W 4W	1. bis 4. (OM 5.1-5.3) (BO) Wiederholung zu SQL → Erweiterung der bisher bekannten SQL Abfragen (Projektion, Selektion, Vereinigung, Join → Übungen zu den bisherigen SQL-Abfragen (Arbeitsblätter) → SQL-Abfragen 2: Vergleichsoperatoren, logische Operatoren, Aggregatfunktionen, mathematische Operatoren → geschachtelte Select-Ausdrücke → Projekt ca. (1) 2 Wochen: Übungen zu allen SQL-Abfragen auf konkreten Datenbanken (Nordwind- und CIA-Datenbank) Software: LibreOffice Writer (BO) 5. bis 9. Wiederholung zum Datenbankentwurf: Grundlagen des ER-Modells (Bezeichnungen und Formen) (OM 3.1.3, 5.3+4-5.3) → Übung zum Entwurf einer ersten Datenbank im ER-Modell → Agile Methoden zur Modellierung von Datenbanken → Modellierungsentscheidungen → Gemischte Übungen zum Datenbankentwurf (inklusive Analyse gegebener ER-Modelle hinsichtlich der Verwendbarkeit in konkreten Sachsituationen und Erweiterungen gegebener Modelle) → Kardinalitäten (1:1, 1:n, n:m) → Rekursive Beziehungen → Mehrfachbeziehungen → Übungen zum Datenbankentwurf mit Kardinalitäten → Umwandlung eines ER-Modells in ein relationales Schema (OM 3.1.3, 5.3+4-5.3) → Übungen zur Umwandlung → Einführung in die Optimierung von Datenbanken (z. B. eine Tabelle mit nicht atomaren Werten, in der bestimmte Filterwünsche nicht vernünftig realisiert werden können) → 1. Normalform (OM 3.1.3, 5.3+4-5.3) → Weitere Probleme an der in die 1. NF umgewandelte Tabelle aufzeigen → 2. Normalform (funktionale Abhängigkeit / voll funktionale Abhängigkeit) (OM 3.1.3, 5.3+4-5.3) → Weitere Redundanzen in der umgewandelten Datenbank aufzeigen → 3. Normalform (transitive Abhängigkeit) (OM 3.1.3, 5.3+4-5.3) → Übungen zur Überführung eines Datenbankschemas in die 1., 2, und 3. Normalform (inklusive Überprüfung, inwiefern ein gegebenes Datenbankschema sich bereits in einer der Normalformen befindet) → Prinzipien des Datenschutzes (BO) PK1.2, PK1.3, PK1.4, PK1.5, PK2.2, PK3.1, PK3.2, IK1.5, IK2.4	- erläutern den Aufbau relationaler Datenbanken unter Verwendung der Begriffe Datensatz, Attribut, Primärschlüssel, Fremdschlüssel und Tabelle. - nennen Beispiele für Einfüge-, Änderungs- und Löschanomalien. - untersuchen ein gegebenes Datenbankschema auf Anomalien und Redundanzen. - formulieren einfache Abfragen und Verbundabfragen über mehrere Tabellen. - formulieren Abfragen an Datenbanken unter Verwendung von Aggregatfunktionen. - interpretieren ein gegebenes ER-Diagramm. - modellieren Datenbanken unter Verwendung des ER-Modells. - setzen ein ER-Modell in ein relationales Schema um. - beurteilen und verändern eine gegebene Datenbankmodellierung.
---	---	--	--

Inhalte Jahrgang 13

Klasse 13				
Inhalte/Themen	Zeit	Hinweise zur Umsetzung		Bezug zum KC
Grundlagen der Theoretischen Informatik 1. Endliche Automaten - DEA, Implementierung DEA , Minimierung DEA, Kellerautomaten DKA , Mealy-Automaten, reguläre Sprachen, reguläre und kontextfreie Grammatiken, NEA , Umwandlung NEA zu DEA	12W	<u>1.</u> Einstieg ins Thema: Smartes Türschloss (Partnerarbeit zu Zuständen, Zustandswechsel und Zustandsgraphen) → Grundbegriffe zu endlichen Automaten (DEA, Eingabealphabet, Ausgabealphabet, Zustandsmenge, Startzustand, Übergangsfunktion, Akzeptor) (OM 5.2.2, 5.4-5.3) (BO) → DEA (Definition) → Projekt ca. 2 Wochen: Implementierung eines DEA als Klasse und switch-case (OM 5.4-5.3) → (optional) Minimierung von DEA (BO) → Mealy-Automat im Kontext Rüsselspringer im Zoo (OM 5.2.2, 5.4-5.3) → Übungen zu MA und DEA (BO) → Grenzen endlicher Automaten (DKA) im Kontext Rüsselspringerfütterautomat mit Übungen → Einführung in reguläre Sprachen („Hawaiianische Sprache“, reguläre Grammatik, Ableitungsbaum) → Übungen zu regulären Sprachen, Grammatiken und Ableitungsbäumen → Zusammenhang DEA und reguläre Grammatiken → DEA zu regulärer Grammatik und reguläre Grammatik zu DEA ableiten → Übungen zum Zusammenhang DEA und regulären Grammatiken → kontextfreie Grammatiken → Übungen zu kontextfreien Grammatiken Einführung in NEA → Übungen zu NEA → Umwandlung NEA in DEA Software: Java-Editor, AtoCC bzw. Flaci (BO) PK1.1, PK1.5, PK2.2, PK2.3, PK3.1, PK3.2 IK3.1, IK3.2, IK3.4 , IK4.3		Theoretische Informatik - beschreiben den Aufbau und die Funktionsweise eines deterministischen endlichen Automaten (DEA). - beschreiben den Aufbau und die Funktionsweise eines endlichen Automaten mit Ausgabe (Mealy-Automat). - entwickeln und implementieren Automatenmodelle in Form von Zustandsgraphen. - analysieren die Funktion eines durch einen Zustandsgraphen vorgegebenen Automaten. - erläutern die Grenzen endlicher Automaten bei der Problemlösung. - beschreiben den Aufbau und die Funktionsweise eines Kellerautomaten als Erweiterung des Modells des endlichen Automaten. - nennen Eigenschaften formaler Sprachen im Vergleich zu natürlichen Sprachen. - beschreiben die von einer Grammatik erzeugte Sprache. - entwerfen reguläre Grammatiken für formale Sprachen. - erläutern den Zusammenhang zwischen regulären Grammatiken und endlichen Automaten.
Künstliche Intelligenz 1. Einstieg: Was ist KI? 2. Maschinelles Lernen - Entscheidungsbäume 3. Reinforcement Learning und Analyse eines automatisierten Lernprozesses	4W	<u>1. bis 3.</u> Einstieg: Akinator → Was ist KI?, Wie funktioniert KI?, Wo findet man KI? → Maschinelles Lernen → Entscheidungsbäume zur Klassifikation von Lebensmitteln und Hunde/Katzen mit Greedy-Algorithmen und GINI-Score → Kann ein Algorithmus rassistisch sein? → Einstieg: Definitionen zum Reinforcement Learning → Drei Spielszenarien zum Reinforcement Learning (Nimm, Frogs and Toads, Labyrinth) → Bekannte KI (OM 1.1-6.1, 1.2-5.3, 2.2-4.2, 3.1-4.3, 4.1-2+4.3, 5.2-6.3, 6.4-5.3)		- diskutieren die Chancen und Risiken der automatisierten Datenanalyse.
Vorschläge für optionale Inhalte für ca. 6 Wochen				
Vertiefungsoption 1: Üben und Wiederholen	Vertiefungsoption 2: Algorithmen und Datenstrukturen 1. Datenstrukturen und abstrakte Datentypen (Liste, Bäume, Graphen) 2. Algorithmische Standardprobleme der Informatik - Suchen (Wegesuche: Tiefen- und Breitensuche in Baumstrukturen und Graphen), Minimierungsprobleme in Graphen (kürzeste Wege mit Dijkstra), Implementierung der oben aufgeführten Algorithmen		Vertiefungsoption 3: Technische Informatik 1. Wahrheitstabellen und Logikgatter (AND, OR, NOT, XOR, NAND, NOR, XNOR) 2. Elementare Schaltnetze (Halb- und Volladdierer bzw. Subtrahierer, Inkrementierer) 3. Schaltnetze mit TTL-Logikgattern und Steckplatinen bauen 4. Minimierung von Schaltnetzen 5. Entwurf komplexerer Schaltnetze 6. Programmierung von Arduinos mit verschiedenen Aktoren und Rezeptoren	Vertiefungsoption 4: theoretische Informatik 1. Minimierung von endlichen Automaten 2. Umwandlung NEA in DEA 3. Turingmaschinen

Leistungsbewertung (Beschluss der Fachkonferenz vom 24.04.2022)

Jahrgänge 9 und 10 (Eine Klassenarbeit pro Halbjahr, 60min, Gewichtung: 40% der Gesamtnote) → Eine schriftliche Leistung kann teilweise oder vollständig aus einem in einer Prüfungssituation erstellten Produkt (Programm, Algorithmus, mit oder ohne externe Hardware,...) bestehen.

Note	6	5	4	3	2	1
ab Prozent	0	20	50	62.5	75	87.5

ab Jahrgang 11 (E-Phase und Q-Phase)

Punkte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ab Prozent	0	20	27	33	40	45	50	55	60	65	70	75	80	85	90	95

Anzahl der Klausuren in der E- und Q-Phase

E-Phase	1.Hj	2.Hj		
Anzahl / Dauer	1 / 90'	1 / 90'		
Gewichtung Leistungen - schriftlich : sonstige	40:60	40:60		
Q-Phase	Q1.1	Q1.2	Q2.1	Q2.2
P4	2 (90')	1 (90')	1 (240')	1 (90')
P5	2 (90')	1 (90')	1 (90')	1 (90')
ohne Prüfung	1 (90')	1 (90')	1 (90')	1 (90')
eA	2 (90')	1 (90')	1 (300')	1 (90')
Gewichtung Leistungen - schriftlich : sonstige	50 : 50 (bei 2 Klausuren und Vorabi) / 40 : 60 (bei 1 Klausur)			

Die Klausuren unter Abiturbedingungen in der Qualifikationsphase sind mit vollständiger Auswahl, wie im schriftlichen Abitur, zu erstellen.