

Schulcurriculum Informatik

Jahrgangsstufen 5 – 10

Madrid, Juli 2025

Gültig ab dem Schuljahr 2025/26

Deutsche Schule Madrid, Fachschaft Informatik
Monasterio de Guadalupe, 7, 28049 Madrid
Tel.: +34 915 580 200
info@dsmadrid.org
www.dsmadrid.org

Das vorliegende Schulcurriculum wurde vom Bund-Länder-Ausschuss für schulische Arbeit im Ausland im September 2025 genehmigt.

Vorwort

Unsere Welt verändert sich rasant. Technologische Entwicklungen wie Digitalisierung, Automatisierung, Künstliche Intelligenz und Datenverarbeitung prägen zunehmend Alltag und Berufswelt. Diese Realität betrifft auch unsere Schülerinnen und Schüler. Als Schule ist es unsere Aufgabe, auf diesen Wandel zu reagieren und unser Bildungsangebot entsprechend weiterzuentwickeln.

Das vorliegende Schulcurriculum für das Fach Informatik an der Deutschen Schule Madrid stellt eine umfassende Überarbeitung bzw. Neugestaltung des im Schuljahr 2021/22 implementierten DSM-Informatik-Lehrplans dar. Es wurde allerdings in zentralen Bereichen aktualisiert und erweitert, um der rasanten technologischen Entwicklung der vergangenen Jahre gerecht zu werden. Besondere Berücksichtigung fanden dabei Fortschritte in den Bereichen Künstliche Intelligenz, maschinelles Lernen, Datenanalyse, Automatisierung und digitale Ethik.

Das Curriculum trägt der Erkenntnis Rechnung, dass Informatikunterricht heute weit mehr ist als das Erlernen von Programmiersprachen: Es geht um das Verstehen digitaler Systeme, den reflektierten Umgang mit Daten und Algorithmen sowie die Fähigkeit, technische Lösungen kreativ, kritisch und verantwortungsvoll mitzugestalten.

Zugleich orientiert sich das Curriculum an den Kompetenzbereichen des Strategiepapiers der Kultusministerkonferenz zur „Bildung in der digitalen Welt“, sodass die Schüler*innen nicht nur technisch, sondern auch methodisch und gesellschaftlich vorbereitet sind, um sich in einer digital geprägten Welt aktiv und sicher zu bewegen.

Besonderes Augenmerk liegt auf der Vermittlung von zukunftsrelevanten Kompetenzen, die sowohl für ein weiterführendes Studium als auch für den Einstieg in eine zunehmend digitalisierte Berufswelt essenziell sind: analytisches Denken, Problemlösefähigkeit, algorithmisches Verständnis, Teamarbeit in technischen Projekten, aber auch digitale Verantwortung und ethische Reflexion.

Mit diesem Curriculum möchten wir unseren Beitrag dazu leisten, die DSM-Schüler*innen nicht nur zu Nutzer*innen digitaler Technologien auszubilden, sondern zu selbstbewussten, kritisch denkenden und verantwortungsvoll handelnden Gestalter*innen unserer digitalen Zukunft.

Inhaltsverzeichnis

1. SEK-I-Informatikunterricht im Zeitalter von KI und Digitalisierung	4
2. Strukturierung des Lehrplans	5
3. Kompetenzen im zeitgemäßen Informatikunterricht	6
4. Fachlehrplan der Jahrgangsstufen 5 bis 10	7
___ Informatik in der Klassenstufe 5	8
___ Informatik in der Klassenstufe 6	12
___ Informatik in der Klassenstufe 7	16
___ Informatik in der Klassenstufe 8	20
___ Informatik in der Klassenstufe 9	25
___ Informatik in der Klassenstufe 10	29
___ Wahlfach Informatik in den Klassenstufen 9 & 10	35
5. Bewertungskriterien und Überprüfbarkeit von Lernleistungen	36

1. SEK-I-Informatikunterricht im Zeitalter von KI und Digitalisierung

Digitale Technologien und Informationssysteme durchdringen heute nahezu alle Lebensbereiche, von Kommunikation und Bildung über Wirtschaft und Wissenschaft bis hin zur Alltagsorganisation. Die fortschreitende Digitalisierung, die globale Vernetzung sowie Entwicklungen wie Automatisierung und Künstliche Intelligenz (KI) eröffnen enorme Chancen, werfen jedoch zugleich ethische, gesellschaftliche und sicherheitsrelevante Fragen auf.

Um diese Veränderungen aktiv und verantwortungsbewusst mitgestalten zu können, benötigen junge Menschen umfassende digitale Kompetenzen. Informatik gilt dabei nicht länger als Zusatzqualifikation, sondern als grundlegende Schlüsselqualifikation („Life Skill“), die sowohl für das Studium und das spätere Berufsleben als auch für die persönliche Urteilsfähigkeit entscheidend ist.

Dabei geht es nicht nur um technisches Know-how: Der Informatikunterricht fördert algorithmisches Denken, Abstraktionsvermögen, kreative Lösungsstrategien und kritische Reflexion. Die Auseinandersetzung mit KI-Systemen – etwa in der Entwicklung von Chatbots oder beim maschinellen Lernen – ist eingebettet in einen breiten Bildungskontext. So lernen die Schüler*innen nicht nur, wie intelligente Systeme funktionieren, sondern auch, welche Auswirkungen sie auf Datenschutz, Meinungsbildung, Arbeitswelt und Verantwortung haben können.

2. Strukturierung des Lehrplans

Der Informatikunterricht lebt von offenen Arbeitsformen und methodischer Vielfalt. Um auf die unterschiedlichen Lernvoraussetzungen einzugehen, sind projektorientierte Phasen fest im Curriculum verankert.

Phase 1: Digitale Grundlagen und Medienkompetenz (Klasse 5 – 7)

In dieser ersten Phase knüpft der Informatikunterricht an die Vorerfahrungen der Schülerinnen und Schüler an, die bereits in der Grundschule mit dem Tablet gearbeitet haben. Diese Kompetenzen werden weiterentwickelt: Sie lernen, digitale Geräte sicher zu nutzen, Informationen gezielt zu recherchieren und sich selbstständig in schulischen Lernumgebungen zurechtzufinden. Dabei stehen Themen wie Dateiverwaltung, Datenschutz und digitale Kommunikation im Mittelpunkt. Erste Erfahrungen im Programmieren, zum Beispiel mit blockbasierten Sprachen, fördern kreatives und strukturiertes Denken.

2. Phase: Programmieren, Vernetzen, Verstehen (Klasse 8 – 10)

In der Mittelstufe steht das kreative und strukturierte Lösen von Problemen im Mittelpunkt. Die Schülerinnen und Schüler entwickeln eigene Programme, arbeiten mit Robotik-Systemen und setzen sich mit objektorientierter Modellierung, Datenverarbeitung und Datenbanken auseinander. Dabei lernen sie sowohl technische als auch gesellschaftliche Aspekte digitaler Systeme kennen, von der Entwicklung intelligenter Anwendungen bis hin zur kritischen Reflexion über Datenschutz, Nachhaltigkeit oder algorithmische Entscheidungsprozesse.

Stundentafel

Die Stundentafel gibt einen Überblick über die Unterrichtsverteilung im Fach Informatik an der DSM-Oberschule:

Jahrgangsstufe	verpflichtend	Informatik Wahlpflichtfach	Informatik Wahlfach
5	1		
6	1		
7	1		
8		4	
9		3	2
10		3	2

3. Kompetenzen im zeitgemäßen Informatikunterricht

Zielsetzung und Bildungsauftrag

Informatik vermittelt Schülerinnen und Schülern zentrale Zukunftskompetenzen für eine digitale Welt. Sie lernen, mit Daten, Algorithmen und digitalen Systemen verantwortungsvoll umzugehen, Prozesse zu analysieren, Probleme modellbasiert zu lösen und technologische Entwicklungen wie Künstliche Intelligenz zu verstehen und kritisch zu reflektieren. Dabei fördert das Fach neben technischem Verständnis auch Kreativität, Teamarbeit, Urteilsfähigkeit und gesellschaftliche Verantwortung.

Zentrale Kompetenzbereiche

- **Analysieren und Modellieren:** Systeme und Problemstellungen strukturieren, abstrahieren und durch geeignete Modelle darstellen.
- **Implementieren und Anwenden:** Lösungen mit passenden Programmiersprachen und Werkzeugen umsetzen, testen und weiterentwickeln.
- **Bewerten und Vernetzen:** Technologische, gesellschaftliche und ethische Auswirkungen einschätzen und informatische Konzepte kontextübergreifend anwenden.
- **Kooperieren und Kommunizieren:** Lösungen im Team entwickeln, fachlich dokumentieren und präsentieren.

Gegenstandsbereiche

Grundlegende Themen wie Algorithmen, Datenstrukturen, Systemarchitekturen, Informationsdarstellung, Modellierung und Softwareentwicklung bilden das fachliche Fundament. Ergänzt werden sie durch gesellschaftlich relevante Fragestellungen wie Datenschutz oder ethische Aspekte Künstlicher Intelligenz. Im Mittelpunkt stehen praxisnahe Anwendungen, projektorientiertes Arbeiten und ein bewusster, reflektierter Umgang mit digitalen Technologien.

4. Fachlehrplan der Jahrgangsstufen 5 bis 10

Der Informatik-Lehrplan der Deutschen Schule Madrid ist spiralcurricular aufgebaut und orientiert sich an einer systematischen, altersgerechten Kompetenzentwicklung. Er begleitet die Schüler*innen kontinuierlich von der 5. Schulstufe bis zum Beginn der Oberstufe und baut schrittweise technische, methodische und gesellschaftlich relevante Fähigkeiten im digitalen Kontext auf.

Im Informatik-Schulcurriculum sind alle sechs Kompetenzbereiche des KMK-Rahmens „Kompetenzen in der digitalen Welt“¹ berücksichtigt. Welche konkreten Teilkompetenzen in welcher Klassenstufe behandelt werden, ist jeweils in der Spalte „KMK“ angegeben.

Die in den Kompetenzformulierungen verwendeten Operatoren entsprechen in ihrer Funktion den Operatoren der KMK und lassen sich den drei Anforderungsbereichen (Reproduktion, Reorganisation/Transfer, Problemlösen/Bewerten) klar zuordnen. Sie ermöglichen eine differenzierte Leistungsbewertung auf allen Anspruchsniveaus – von der Wiedergabe grundlegenden Fachwissens bis zur selbstständigen Problemlösung und kritischen Reflexion. Für das Fach Informatik können dabei vereinzelt Operatoren verwendet werden, die nicht eins zu eins den Formulierungen der KMK entsprechen, was den fachlichen Besonderheiten geschuldet ist.

Hinweis: Im folgenden Informatik-Curriculum sind projektorientierte Arbeiten einzelnen Lernbereichen zugeordnet, um eine sinnvolle inhaltliche Verankerung und Orientierung zu bieten. Diese Zuordnung stellt jedoch eine Empfehlung dar und ist nicht verpflichtend: Lehrkräfte haben die pädagogische Freiheit, Projekte auch mit anderen Lernbereichen zu verknüpfen, sofern dies dem Lernprozess und den Zielkompetenzen dient.

¹ KMK Kultusminister Konferenz (2017): „Bildung in der digitalen Welt. Strategie der Kultusministerkonferenz. Online verfügbar unter: <https://www.kmk.org/themen/bildung-in-der-digitalen-welt/strategie-bildung-in-der-digitalen-welt.html> (abgerufen am 03.06.2025).

Informatik in der Klassenstufe 5

Einstieg in die digitale Lernumgebung

Der Informatikunterricht in Klasse 5 knüpft an die bereits in der Grundschule gesammelten Erfahrungen mit dem Tablet an und dient der vertiefenden Einführung in die digitale Schulumgebung. Die Schülerinnen und Schüler lernen, digitale Arbeitsplätze sicher und zunehmend selbstständig zu nutzen. Dabei stehen grundlegende Kompetenzen im Umgang mit Informatiksystemen, digitalen Daten und Kommunikationsdiensten im Vordergrund. Der Unterricht ist eng mit dem Deutschunterricht verknüpft und fördert bewusst fächerübergreifendes Arbeiten.

Im ersten Lernbereich erweitern die Schülerinnen und Schüler ihre Kenntnisse in der Daten- und Benutzerverwaltung. Sie lernen, Dateien strukturiert zu speichern, lokale Ordnerhierarchien sinnvoll zu nutzen und ergänzend erste Erfahrungen mit digitalen Speichermedien zu sammeln. Die Einführung in cloudbasierte Ablagesysteme erfolgt aufbauend nach dieser Basisschulung.

Im zweiten Lernbereich stehen der sichere und respektvolle Umgang mit digitalen Medien und Kommunikationsdiensten im Fokus. Die Lernenden reflektieren Risiken im Netz und erhalten altersgerechte Einblicke in Datenschutz, Urheberrecht und digitale Kommunikation.

Den Abschluss bildet eine projektorientierte Arbeit, bei der die Schülerinnen und Schüler eine multimediale Präsentation zu einem selbstgewählten Thema gestalten – unter Einbezug von Bildern, strukturierter Internetrecherche und korrekter Quellenangabe. Das Projekt verbindet Inhalte aus Informatik und Deutsch und stärkt sowohl die Präsentations- als auch die Medienkompetenz.

Lernbereich 1: Digitale Grundkompetenz (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Technische Grundverständnis - Aufbau moderner digitaler Geräte (PC, Tablet, Smartphone) und Unterscheidung von Ein-/Ausgabegeräten - Grundbegriffe: Hardware, Software, Betriebssystem - Prinzip der Datenverarbeitung (Eingabe – Verarbeitung – Ausgabe – Speicherung) Dateiorganisation und Speichermedien - Nutzung lokaler und cloudbasierter Speicherlösungen - Organisieren und Strukturieren von Dateien und Ordern im Dateisystem - Pfadangaben verstehen und anwenden (lokal & online) Bedienkompetenz - Einführung in Shortcuts, ergonomisches Tippen (z. B. 10-Finger-System) Datenschutz und Sicherheit - Schutz persönlicher Daten und sicheres Verhalten in digitalen Umgebungen	Technisches Grundverständnis: - benennen zentrale Hardwarekomponenten (z. B. Prozessor, Arbeitsspeicher, Festplatte, Ein-/Ausgabegeräte) und beschreiben deren grundlegende Funktion - unterscheiden zwischen Hardware, Software und Betriebssystem und können diese Begriffe an Beispielen erklären. - erläutern das EVA-Prinzip anhand einfacher Beispiele aus dem Alltag oder dem Unterricht Dateiorganisation und Speichermedien - organisieren Dateien und Ordner sinnvoll in einer hierarchischen Struktur und erstellen einfache Ordnerbäume - verstehen und verwenden Pfadangaben, um Dateien lokal und online gezielt zu finden - verwenden lokale sowie cloudbasierte Speicherlösungen (z. B. OneDrive) zur Ablage, Wiederverwendung und Sicherung von digitalen Inhalten - verwalten Dateien eigenständig (z. B. Verschieben, Umbenennen, Löschen) und nutzen grundlegende Dateimanagementfunktionen sicher Bedienkompetenz - wenden grundlegende Shortcuts (z. B. Strg+C, Strg+V, Strg+Z) zur effizienten Arbeit am Computer an - wenden grundlegende ergonomische Schreibtechniken an, z. B. das 10-Finger-Tastschreiben in Übungsphasen oder mit Lernprogrammen	5.1.2 5.1.1 5.5.1 1.3.2 1.3.1 1.3.1 1.3.2 5.2.1 4.3.2

Lernbereich 2: Medien und Kommunikation – Sicherheit im Netz (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Datenschutz und Sicherheit • Passwortschutz & Identität: Erstellung sicherer Passwörter, Schutz persönlicher Daten, bewusster Umgang mit Online-Profilen (z. B. in Lernplattformen) • Gefahren erkennen: Authentizität von Nachrichten beurteilen (Phishing, Fake-Absender), einfache Sicherheitsstrategien anwenden Grundlagen Internet & Kommunikation • Kommunikation per E-Mail: Erstellen, beantworten, weiterleiten von Nachrichten mit Betreff, Anhang und korrekter Ansprache • Dateianhänge & Medienkompetenz: Geeigneter Umgang mit Anhängen, Dateigröße & -typ kennen, Urheberrecht beachten • KI zur Korrektur und Überarbeitung von Texten benutzen • Netiquette & Datenschutz: Einhaltung respektvoller Kommunikationsregeln, Schutz eigener und fremder Daten	Datenschutz und Sicherheit • erstellen sichere Passwörter unter Berücksichtigung gängiger Sicherheitskriterien. • schützen persönliche Daten durch einen bewussten Umgang mit Online-Profilen, insbesondere auf Lernplattformen. • beurteilen die Authentizität digitaler Nachrichten zur Erkennung von Phishing und gefälschten Absendern. • wenden einfache Sicherheitsstrategien wie Zwei-Faktor-Authentifizierung und Passwortmanagement zur Erhöhung der digitalen Sicherheit an. Grundlagen Internet & Kommunikation • verfassen E-Mails mit eindeutigem Betreff, passender Anrede und klarer Struktur. • beantworten und leiten E-Mails sachgerecht weiter unter Einhaltung kommunikativer und formaler Standards. • verwenden Dateianhänge unter Berücksichtigung von Format, Dateigröße und Zweckmäßigkeit. • beachten urheberrechtliche Vorgaben beim Einsatz und der Weitergabe von Dateien und digitalen Medien. • halten respektvolle Kommunikationsregeln nach den Prinzipien der Netiquette im digitalen Austausch ein. • berücksichtigen Datenschutzaspekte beim Kommunizieren und Teilen persönlicher sowie fremder Informationen • überarbeiten Texte mithilfe KI-gestützter Werkzeuge und bewerten Vorschläge hinsichtlich sprachlicher Qualität, Angemessenheit und Verlässlichkeit	4.1.2 4.2.2 4.1.1 4.2.1 2.1.1 2.1.2 2.2.1 3.3.2 2.4.1 4.2.1 3.2.2 1.2.1 6.2.2

Lernbereich 3: Projekt – Digitale Präsentation (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Recherche & Medienauswahl • zielgerichtete Bildrecherche mit Nutzung von Filterfunktionen (z. B. nach Größe, Lizenz, Farbe) • Bewertung von Bildern nach inhaltlichen, gestalterischen und technischen Kriterien • Auswahl und Verwendung frei lizenzierter Materialien (z. B. Creative Commons) unter Berücksichtigung des Urheberrechts und der Quellenangabe Erstellen digitaler Präsentationen • Planung und Gestaltung klar strukturierter Präsentationen zu fachlichen Themen (z. B. aus dem Deutschunterricht) • Fokus auf Lesbarkeit, durchgängige Gestaltung (Farben, Schriftarten, Layout) und Reduktion auf zentrale Inhalte • Anpassung der Präsentationsweise an Zielgruppe und Präsentationssituation Medienbearbeitung & Integration • Einfügen und technisch sauberes Anpassen von Bildern (z. B. Skalierung, Kontrast, Seitenverhältnis) • gezielter Einsatz von Layouts, Farbkonzepten, Schriften und einfachen Animationen zur Unterstützung der Aussage • Kombination von Text-, Bild- und Gestaltungselementen zu einem kohärenten Gesamteindruck • KI-generierte Bilder in einer sicheren KI (z.B. Fobizz) abspeichern und benutzen, problematisieren der Quelle (Datenschutz, Copyright etc.) Quellenangabe & Veröffentlichung: • korrekte und transparente Zitation von Bildquellen, insbesondere bei der Nutzung fremder Materialien • datenschutzkonforme Veröffentlichung oder Weitergabe digitaler Präsentationen über schulische Plattformen oder Cloud-Dienste (z. B. OneDrive, Schul-Cloud) • Bewusstsein für Verantwortung beim digitalen Publizieren (Rechte Dritter, Formatwahl)	Recherche & Medienauswahl • recherchieren zielgerichtet Bilder mithilfe von Filterfunktionen (z. B. nach Größe, Lizenz oder Farbe). • bewerten Bilder nach inhaltlichen, gestalterischen und technischen Kriterien. • wählen frei lizenzierte Materialien aus und verwenden diese unter Berücksichtigung des Urheberrechts und der korrekten Quellenangabe. Erstellen digitaler Präsentationen • planen und gestalten klar strukturierte Präsentationen zu fachlich vorgegebenen Themen • fokussieren Inhalte auf zentrale Aussagen und sichern eine konsistente Gestaltung hinsichtlich Farben, Schriften und Layout. • passen die Präsentationsweise an Zielgruppe und Präsentationssituation an Medienbearbeitung & Integration • fügen Bilder in digitale Produkte ein und passen sie technisch korrekt an (z. B. Skalierung, Kontrast, Seitenverhältnis) • setzen Layouts, Farben, Schriften und Animationen gezielt zur inhaltlichen Unterstützung ein • kombinieren Text-, Bild- und Gestaltungselemente zu einem kohärenten Gesamteindruck • verwenden KI-generierte Bilder in sicheren Tools (z. B. Fobizz), reflektieren deren Herkunft und Energieverbrauch kritisch und berücksichtigen Datenschutz sowie urheberrechtliche Aspekte bei der Weiterverwendung Quellenangabe & Veröffentlichung • zitieren Bildquellen korrekt und transparent, insbesondere bei der Nutzung fremder Materialien. • veröffentlichen oder teilen digitale Präsentationen datenschutzkonform über schulische Plattformen oder Cloud-Dienste. • reflektieren die Verantwortung beim digitalen Publizieren in Bezug auf Urheberrechte, Zielgruppe und Formatwahl.	1.1.2 1.2.1 3.3.2 3.1.2 3.2.1 2.4.2 3.2.1 3.1.2 3.2.2 3.3.2 4.4.1 2.2.2 3.3.2 3.1.2 4.2.1 3.3.1 6.2.2

Informatik in der Klassenstufe 6

Informieren, Programmieren, Gestalten

In Klasse 6 vertiefen die Schülerinnen und Schüler ihre digitale Handlungskompetenz in drei zentralen Bereichen: Informationsverarbeitung, algorithmisches Denken und Textgestaltung.

Im Umgang mit digitalen Medien lernen sie, gezielt zu suchen, Informationen kritisch zu bewerten und dabei rechtliche Rahmenbedingungen wie Urheberrecht und Lizenzen zu beachten. Sie organisieren Inhalte strukturiert und reflektieren den sicheren Umgang mit persönlichen Daten – auch im Team und in schulischen Plattformen.

Im Bereich der Algorithmik gestalten sie einfache interaktive Anwendungen, insbesondere Spiele, in einer blockbasierten Programmierumgebung. Dabei setzen sie grundlegende Steuerungsstrukturen wie Verzweigungen und Schleifen ein. Das eigenständige Planen, Testen und Optimieren fördert logisches Denken und Kreativität.

Abschließend gestalten die Schülerinnen und Schüler in einem digitalen Projekt längere Texte unter Berücksichtigung von formalen und gestalterischen Standards. Sie nutzen ein Textverarbeitungsprogramm gezielt zur strukturierten Aufbereitung von Inhalten, beispielsweise im Kontext des Deutschunterrichts.

Lernbereich 1: Medien und Kommunikation - digitale Kommunikationsverarbeitung (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Suchmaschinen • verschiedene Suchmaschinen kennen und vergleichen (z. B. Google, Bing, DuckDuckGo, You.com) • Informationsquellen gezielt einsetzen (z. B. Enzyklopädien, Datenbanken, schulische Plattformen) • KI-gestützte Suchtools und Chatbots (z. B. ChatGPT, Perplexity) sinnvoll in den Rechercheprozess einbinden	Suchmaschinen • recherchieren gezielt mit verschiedenen Tools und Plattformen, einschließlich KI-gestützter Systeme • wählen geeignete Quellen aus und bewerten Informationen kritisch nach Relevanz, Herkunft und Absicht • vergleichen klassische und KI-gestützte Suchprozesse hinsichtlich Qualität, Transparenz und Ergebnisdarstellung	1.1.2 1.2.1 1.2.2 6.2.2
Vernetzte Strukturen • vernetzte Informationsnetzwerke aus Alltag und Schule analysieren (z. B. Schulwebsite, Online-Nachschlagewerke) • Hypertextstrukturen erkennen und modellieren (z. B. mit Knoten-Kanten-Modellen)	Vernetzte Strukturen • analysieren digitale Informationsstrukturen im Alltag und Schulkontext • modellieren einfache vernetzte Strukturen mit geeigneten Darstellungen (z. B. Hypertextmodelle, Graphen) • reflektieren Aufbau, Navigation und Orientierungsmöglichkeiten in digitalen Informationssystemen	6.1.1 5.5.3 6.2.1 6.2.2
Digitale Inhalte erstellen • einfache verlinkte Inhalte gestalten (z. B. Mini-Websites, Wiki-Einträge) • Quellen korrekt angeben (Autor, URL, Lizenz, ggf. Chatbot-Hinweis bei KI-Nutzung)	Digitale Inhalte erstellen • erstellen digitale Inhalte mit interner Verlinkung, strukturierter Navigation und übersichtlichem Aufbau • zitieren genutzte Quellen transparent und urheberrechtskonform, inklusive KI-basierter Quellenangaben	3.1.2 2.2.2 3.3.2
Grundlagen der Internetstruktur • grundlegende Struktur und Funktionsweise des Internets beschreiben (Client-Server, Router, Dienste wie WWW) • Adressierung und Zugriff im Internet verstehen (z. B. URL, IP-Adressen, DNS) • Unterschiede zwischen relativen und absoluten Pfadangaben klären	Grundlagen der Internetstruktur • beschreiben zentrale Komponenten und Funktionsweisen der Internetinfrastruktur • erklären Prinzipien der Adressierung und Datenweiterleitung im Netz • unterscheiden verschiedene Pfadarten in digitalen Projekten und wenden diese kontextgerecht an	5.5.1 5.5.1 1.3.1 5.2.4
	Optionale Vertiefung • entwickeln ein schulisches Wiki mit interner Verlinkung, thematischem Fokus (z. B. „Naturwissenschaften“) und sauberer Quellenarbeit • verwenden Chatbots zur Vorstrukturierung von Informationen und reflektieren deren Vorteile und Grenzen gegenüber klassischen Suchwegen	3.1.2 2.2.2 3.3.2 1.1.2 6.2.2

Lernbereich 2: Algorithmik – Entscheidungen treffen und Spiele gestalten (12 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Daten verarbeiten • Verwendung von Variablen zur Speicherung und Weiterverarbeitung von Daten • Ein- und Ausgabe von Werten in digitalen Anwendungen (z. B. Text, Zahlen) • einfache Berechnungen und Zuweisungen mit Variablen Algorithmische Strukturen verwenden • Verwendung von Bedingungen mit logischen und Vergleichsoperatoren • einfache Verzweigungen mit if/else-Strukturen • Einsatz von Schleifen (z. B. kopfgesteuert oder zählgesteuert) Algorithmen analysieren und testen • gezieltes Testen von Programmen zur Überprüfung auf Korrektheit • Identifikation und Behebung von Fehlern (Debugging) • Kontrolle von Variablenwerten und Programmezuständen während der Ausführung Anwendungsbezug: Spieleentwicklung • Gestaltung eines interaktiven Spiels mit einer blockbasierten Programmiersprache (z. B. Scratch) • Umsetzung algorithmischer Konzepte wie Bedingungen, Wiederholungen und Zufall • Erweiterung durch spieltypische Elemente wie Punktestand, Leben oder Zeitsteuerung	Daten verarbeiten • verarbeiten Daten mit Hilfe von Variablen und einfachen Ausdrücken • unterscheiden zwischen Eingabe, Verarbeitung und Ausgabe in algorithmischen Abläufen • wenden Variablen zur dynamischen Steuerung von Programmprozessen an Algorithmische Strukturen verwenden • steuern Programmläufe durch Bedingungen und Wiederholungen • wählen geeignete Kontrollstrukturen für konkrete Aufgabenstellungen aus • anwenden logische Operatoren zur Entscheidungsfindung in interaktiven Abläufen Algorithmen analysieren und testen • analysieren Programme durch systematische Tests und Datenauswertung • erkennen typische Fehlerquellen und beheben sie mithilfe einfacher Debugging-Strategien • kontrollieren Zwischenergebnisse zur Sicherstellung der Programmlogik Anwendungsbezug: Spielentwicklung • entwickeln ein kreatives Spielprojekt mit iterativen Verbesserungsprozessen • kombinieren verschiedene algorithmische Konzepte zur Steuerung des Spielverhaltens • reflektieren Spielmechaniken und optimieren Abläufe zur Verbesserung der Nutzererfahrung Vorschläge zur Vertiefung: • verwenden verschachtelte Bedingungen zur Umsetzung komplexerer Entscheidungslogiken • integrieren erweiterte Spielmechaniken wie Punktestand, Lebenssysteme oder Zeitlimits • testen Feedbackschleifen zur Optimierung von Spielbalance und Nutzerinteraktion	5.5.2 5.5.1 5.5.3 5.5.3 5.5.2 5.5.3 5.1.2 5.1.3 5.5.3 5.1.3 3.1.2 5.5.3 6.2.2 5.5.3 5.5.2 5.1.2 5.1.3

Lernbereich 3: Projekt – Texte gestalten mit digitalen Werkzeugen (8 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Texte strukturieren und überarbeiten - digitale Texte überarbeiten mit Blick auf Sprache, Klarheit und Struktur - Absätze, Zwischenüberschriften und Seitenumbrüche gezielt einsetzen, um Leseführung und Übersicht zu verbessern - KI-gestützte Schreib- und Korrekturhilfen (z. B. in Word, Google Docs oder KI-Tools wie Grammarly, ChatGPT) nutzen und kritisch reflektieren Layout und Format gezielt einsetzen - Zeichen- und Absatzformate gezielt anwenden (z. B. Schriftarten, Ausrichtung, Zeilenabstand, Einzüge) - Formatvorlagen verwenden und anpassen, um einheitliche Layouts für Präsentationen, Berichte oder Projektarbeiten zu gestalten - moderne Layoutfunktionen wie Live-Vorschau, Design-Ideen (z. B. in Microsoft 365) oder dynamische Formatierungseffekte sinnvoll einsetzen Medien wirkungsvoll integrieren - Bilder in digitale Texte einfügen und passend zur Aussage und Leseführung positionieren - Anpassung von Größe, Textumfluss, Seitenverhältnis und ggf. barrierefreier Beschreibung (Alt-Text) - Nutzung lizenzfreier Bilddatenbanken und KI-generierter Bilder (mit Kennzeichnung) Rechte klären, Quellen korrekt angeben - korrekte Zitation von Text- und Bildquellen, inkl. Verweis auf KI-gestützte Textgeneratoren (z. B. „erstellt mit Unterstützung durch ChatGPT“) - Beachtung von Datenschutz, insbesondere bei personenbezogenen Inhalten in Texten und Bildern - Verantwortungsvoller Umgang mit fremden Inhalten bei der Veröffentlichung oder Präsentation (z. B. im Schulnetz oder im Web)	Texte strukturieren und überarbeiten - strukturieren digitale Texte sinnvoll unter Beachtung sprachlicher Korrektheit und klarer Gliederung - überarbeiten Texte unter Einbezug digitaler Werkzeuge zur Korrektur und Verbesserung - nutzen automatische Hilfen reflektiert und bewerten deren Vorschläge kritisch Layout und Format gezielt einsetzen - formatieren Texte konsistent und zweckorientiert mit passenden Layoutmerkmalen - verwenden Formatvorlagen zur Strukturierung umfangreicher Dokumente - reflektieren den gestalterischen und funktionalen Einsatz typografischer Mittel Medien wirkungsvoll integrieren - integrieren Bilder funktional in Textdokumente und passen sie technisch korrekt an - beurteilen die Wirkung von Bildern hinsichtlich Inhalt, Relevanz und Gestaltung - ergänzen Texte durch mediale Elemente zur inhaltlichen und ästhetischen Aufwertung Rechte klären, Quellen korrekt angeben - zitieren Quellen korrekt, transparent und formatgerecht - beachten rechtliche Rahmenbedingungen zu Urheberrecht, Lizenzen und Datenschutz - reflektieren den Umgang mit KI-generierten Inhalten im Kontext von Urheberschaft und Transparenz Optionale Vertiefung - bearbeiten und individualisieren bestehende Formatvorlagen (z. B. für eine Projektarbeit, Bewerbung oder digitale Broschüre) - exportieren Texte in verschiedene Formate (PDF, DOCX, HTML) zur Weitergabe oder digitalen Präsentation - wenden cloudbasierte Tools (z. B. OneDrive) zur kollaborativen Textgestaltung in Echtzeit - erproben plattformübergreifende Darstellung von Dokumenten	3.1.2 5.2.1 5.4.1 6.2.2 3.1.2 3.2.2 6.1.1 3.2.1 6.1.1 3.1.2 2.2.2 3.3.2 4.2.1 3.3.1 6.2.2 3.2.2 3.1.2 2.3.2 5.2.4

Informatik in der Klassenstufe 7

Digitale Grundlagen im Kontext von Algorithmik, Kommunikation und KI

Die Schüler*innen entwickeln algorithmisches Denken durch das systematische Lösen konkreter Probleme. Sie formulieren Abläufe als Algorithmen und nutzen blockbasierte Programmierung als Werkzeug. Im Fokus steht dabei nicht die Programmiersprache, sondern das strukturierte Denken in Abläufen, Bedingungen und Wiederholungen.

Beim Gestalten interaktiver Programme modellieren sie einfache Szenarien, setzen Kontrollstrukturen gezielt ein und dokumentieren ihre Arbeit mit digitalen und multimedialen Mitteln. Erste KI-nahe Konzepte wie Zufall, Mustererkennung und regelbasierte Entscheidungen fließen dabei mit ein.

Abschließend analysieren sie größere Datenmengen mithilfe von Tabellenkalkulationen. Sie filtern, visualisieren und interpretieren Daten und schaffen damit die Grundlage für datenbasierte Entscheidungen – auch im Hinblick auf datengetriebene Prozesse in Künstlicher Intelligenz und Data Science.

Lernbereich 1: Algorithmik – digitale Strukturen verstehen und gestalten (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Algorithmisches Denken & Datenkompetenz • Abläufe erkennen, modellieren und bewerten – z. B. Regeln, Szenarien oder Prozessschritte • Entwicklung interaktiver Programme mit menschenähnlichen Programmiersprachen wie Python • Anwendung von verschachtelten Strukturen, Bedingungen und Kontrollflüssen • Unterscheidung und Nutzung verschiedener Datentypen (Zahlen, Texte, Wahrheitswerte) • gezielter Einsatz von Operatoren zur Datenverarbeitung • Programmsteuerung durch Bedingungen (ein- und zweiseitig) sowie Wiederholungen Digitale Medien kreativ gestalten • Gestaltung interaktiver digitaler Medien, z. B. Anwendungen, Animationen oder Programme mit Benutzerinteraktion • Erstellung eigener Grafiken (Raster- oder Vektorbilder) für digitale Produkte • multimediale Dokumentation von Arbeitsprozessen und Ergebnissen mit Video, Audio, Screenshots oder Tutorials • Berücksichtigung von Urheber- und Persönlichkeitsrechten bei der Nutzung und Veröffentlichung Medienkompetenz & digitale Verantwortung • Funktionsweisen digitaler Programme mit spielerischen Elementen untersuchen, z. B. Punktesysteme, Belohnungen, Fortschrittsanzeigen oder Rückmeldungen in Lern-Apps und Online-Plattformen • kritische Reflexion der eigenen Mediennutzung und digitaler Gewohnheiten • Grundlagen sicherer Kommunikation: Datenschutz, Passwörter, Verschlüsselung • gesundheitliche Aspekte der Mediennutzung: Bildschirmzeiten, Selbstregulation, digitale Balance	Algorithmisches Denken & Datenkompetenz • entwickeln und beschreiben algorithmische Abläufe mit menschenähnlicher Syntax wie Python • identifizieren Datentypen und wenden passende Operationen kontextbezogen an • wenden Kontrollstrukturen wie Bedingungen und Schleifen zur Steuerung interaktiver Programme an Digitale Medien kreativ gestalten • gestalten eigene interaktive Programme unter Einbindung selbst erstellter grafischer Elemente • dokumentieren Arbeitsschritte und Ergebnisse kreativ mit geeigneten Medienformaten • beachten rechtliche Vorgaben bei der Erstellung und Veröffentlichung multimedialer Inhalte Medienkompetenz & digitale Verantwortung • analysieren Funktionsweisen und Wirkmechanismen digitaler Systeme mit spielerischem Charakter • reflektieren eigene digitale Routinen und bewerten mögliche Auswirkungen auf Konzentration und Wohlbefinden • kommunizieren sicher und verantwortungsvoll unter Berücksichtigung digitaler Selbstschutzmaßnahmen	 5.5.3 5.5.2 5.5.3 3.1.2 3.2.2 3.3.2 6.1.3 5.5.1 6.2.2 4.3.2 4.1.2 2.1.2

Lernbereich 2: Medien und Kommunikation – Datensicherheit (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Grundlagen der Verschlüsselung • einfache Verschlüsselungsverfahren verstehen und selbst anwenden (z. B. Caesar-Code) • Begriffe wie Klartext, Geheimtext, Schlüssel erklären • Bedeutung vertraulicher Kommunikation erkennen Sicher kommunizieren im Netz • Verschlüsselung im Alltag erkennen (https, sichere WLANs, E-Mail) • starke Passwörter entwickeln – technisch sinnvoll und leicht zu merken • Dateien verschlüsseln und geschützt verschicken Umgang mit Daten und digitalen Risiken • Unterschiede zwischen sicherer und unsicherer Kommunikation erkennen • Risiken wie Tracking, Fake News oder Manipulation durchschauen • eigenes Verhalten im Netz kritisch hinterfragen	Grundlagen der Verschlüsselung • erläutern Grundprinzipien von Verschlüsselung mit anschaulichen Beispielen • bewerten einfache Verfahren auf ihre Sicherheit • verschlüsseln kurze Nachrichten praktisch und tauschen sie sicher aus Sicher kommunizieren im Netz • erkennen Verschlüsselung im Alltag und nutzen sie gezielt • erstellen sichere Passwörter und erklären deren Aufbau • versenden Daten geschützt mit einfachen Tools Umgang mit Daten und digitalen Risiken • unterscheiden sichere von unsicheren Kommunikationswegen • reflektieren die Nutzung digitaler Dienste und bewerten Risiken • beurteilen den Umgang mit persönlichen Daten in digitalen Medien	 5.5.1 4.1.1 4.1.2 4.2.3 4.1.2 4.1.2 4.2.1 4.1.1 6.2.2 4.2.2 6.2.2

Lernbereich 3: Projekt – Tabellenkalkulation und Datenanalyse (10 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Grundlagen der Tabellenkalkulation - Aufbau und Struktur digitaler Tabellen verstehen (Tabellenblatt, Zelle, Zeile, Spalte, Zelladresse) - bestehende Tabellen analysieren, Daten eintragen und zwischen Zahlen und Textwerten unterscheiden - Formatierungen gezielt einsetzen (z. B. Währung, Dezimalstellen, Prozentwerte)	Grundlagen der Tabellenkalkulation - verwenden grundlegende Tabellenbegriffe sicher und kontextgerecht - ergänzen Daten strukturiert und unterscheiden zwischen Zahlen- und Textformaten - formatieren Tabelleninhalte übersichtlich und leserfreundlich	5.2.1 1.3.2 3.1.2
Datenauswertung und Visualisierung - Tabellen gezielt planen und strukturieren, inklusive klarer Bezeichnungen und einheitlicher Eingaben (z. B. Codierungen, Dropdowns) - verschiedene Diagrammtypen mit Assistenten erstellen (Balken, Säulen, x-y-Diagramm) - Diagramme beschriften, automatisch aktualisieren und auf Verständlichkeit prüfen	Datenauswertung und Visualisierung - strukturieren Datentabellen mit klaren Angaben und konsistenter Eingabelogik - erstellen passende Diagramme zur Datenvisualisierung und erklären deren Aufbau - bewerten die Aussagekraft und Verständlichkeit von Visualisierungen	1.3.2 3.1.2 1.2.1 6.1.1
Formeln und Automatisierung - einfache Formeln mit Grundrechenarten entwickeln (z. B. Summen, Differenzen, Mittelwerte) - Verwendung relativer Zellbezüge zur automatisierten Berechnung - Vorteile automatisierter Tabellenabläufe erkennen (z. B. bei Datenänderungen oder Wiederholungen)	Formeln und Automatisierung - entwickeln eigene Berechnungen mit Hilfe einfacher Formeln - wenden relative Zellbezüge zur flexiblen Datenverarbeitung an - reflektieren den Nutzen automatisierter Abläufe bei größeren Datenmengen	5.5.3 5.5.2 6.2.2
Vertiefung - Anwendung erweiterter Formatierungen (z. B. benutzerdefinierte Datums- oder Zahlenformate, bedingte Formatierung) - Gestaltung professioneller Tabellen für Projekte oder Präsentationen	Vertiefung - wenden weiterführende Formatierungen für anspruchsvollere Anwendungsfälle an - gestalten Tabellen mediengerecht für verschiedene Ausgabesituationen	3.2.2 3.1.2

Informatik in der Klassenstufe 8

Kreativ programmieren, vernetzt denken, gemeinsam lösen

In der 8. Klasse Informatik erwerben die Schüler*innen grundlegende Kompetenzen im Umgang mit digitalen Technologien, Robotik und Programmierung. Beim Planen, Bauen und Programmieren von Robotern verbinden sie technisches Verständnis mit Kreativität, Teamarbeit und algorithmischem Denken. Sie lernen, wie Sensoren und Aktoren zusammenwirken, entwickeln Steuerungslogiken und setzen eigene Ideen in funktionsfähige Systeme um. Dabei erfahren sie, wie Hard- und Software interagieren, und reflektieren über gesellschaftliche Aspekte wie Automatisierung, Nachhaltigkeit oder künstliche Intelligenz.

Gleichzeitig vertiefen sie ihr Verständnis für Programmierung als Methode der Problemlösung. Sie arbeiten mit prozeduralen Sprachen, nutzen Kontrollstrukturen, Variablen, Schleifen und Funktionen, um Abläufe systematisch zu modellieren und umzusetzen. Auch das Lesen, Testen und Optimieren von Code sowie der Einsatz grafischer Planungsmittel gehören dazu.

In einem abschließenden Projekt wenden die Schüler*innen ihr Wissen praxisnah an: Sie entwickeln im Team ein eigenes Programm oder Robotiksystem, dokumentieren den Entwicklungsprozess und präsentieren ihre Ergebnisse. Dabei stärken sie ihre Selbstlern-, Kommunikations- und Gestaltungskompetenz in einer zunehmend digitalen Welt.

Lernbereich 1: Robotik und digitale Vernetzung (40 UE)

Beim Bauen und Programmieren von Robotern verbinden die Schüler*innen technische, mathematische und algorithmische Denkweisen mit Kreativität und Teamarbeit. Sie lernen, wie Sensoren und Aktoren über definierte Schnittstellen miteinander kommunizieren und wie diese gezielt in Steuerungslogiken eingebunden werden. Dabei wird ein grundlegendes Verständnis für vernetzte Systeme und die Interaktion zwischen Hard- und Software entwickelt.

Beim Arbeiten an eigenen Robotik-Projekten erkennen die Schüler*innen fachübergreifende Zusammenhänge und reflektieren die Bedeutung digitaler Technologien für Gesellschaft und Zukunft. Neben technischem Verständnis fördern die Projekte Problemlösefähigkeit, kritisches Denken und Teamarbeit – zentrale Kompetenzen in einer digitalen Welt.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Digitale Systeme - Simulation einfacher Abläufe maschinennaher Programmierung auf Basis eines vereinfachten von-Neumann-Modells - Gegenüberstellung maschinennaher und höherer Programmiersprachen hinsichtlich Effizienz und Verständlichkeit - Veranschaulichung digitaler Repräsentation durch Binärsysteme, Ganzzahlen, Dezimalzahlen - Thematisierung technischer Grenzen wie Rundungsfehler und Wertebereichsüberschreitungen - Einführung in logische Verknüpfungen und einfache binäre Additionen Reflexion über die Rolle binärer Systeme in heutigen digitalen Anwendungen Robotik - Bau, Konfiguration und Inbetriebnahme eines autonomen Roboters (z. B. LEGO Mindstorms oder kompatible Systeme) - Laden, Verändern und Testen von Programmen auf Robotersystemen Verwendung verschiedener Sensoren (z. B. Licht-, Berührungs-, Ultraschallsensor) mit praxisnaher Parametrierung - Programmierung mit Kontrollstrukturen: Schleifen (z. B. „wiederhole“, „wiederhole bis“), Bedingungen (if, if/else) - Nutzung von Variablen, Zufallswerten und Vergleichsoperatoren zur dynamischen Steuerung - Modellierung von Roboterverhalten mit Zustandsdiagrammen Umsetzung einer Teamaufgabe wie z. B. ein Roboter-Wettbewerb mit algorithmischem Ablauf - kritische Auseinandersetzung mit Chancen und Risiken digitaler Technologien (z. B. Smart Home, KI, Nachhaltigkeit, Datenschutz)	Digitale Systeme - analysieren die grundlegende Funktionsweise eines Informatiksystems anhand eines vereinfachten von-Neumann-Modells - erklären den Unterschied zwischen maschinennaher und höherer Programmierung mit Blick auf Effizienz und Anwendbarkeit - erläutern die Grenzen digitaler Darstellung anhand technischer Beispiele wie Rundungsfehlern oder Wertebereichsüberschreitungen - führen einfache binäre Rechenoperationen durch und interpretieren deren Bedeutung für die maschinelle Verarbeitung Robotik - bauen, konfigurieren und testen autonome Roboter unter Berücksichtigung von Hardware- und Softwareaspekten - verwenden Sensoren gezielt zur Erfassung von Umweltinformationen und steuern deren Auswertung in Programmen - entwickeln Programme mit strukturierten Kontrollflüssen (z. B. Schleifen, Bedingungen) zur Lösung komplexer Steuerungsaufgaben - modellieren Roboterverhalten mit Zustandsdiagrammen und setzen diese in funktionsfähige Programme um - bewerten Programmstrukturen hinsichtlich ihrer Effizienz, Funktionalität und Erweiterbarkeit - analysieren Chancen und Risiken aktueller Technologien unter Aspekten wie Nachhaltigkeit, Datenschutz und gesellschaftlicher Relevanz	5.5.1 5.5.1 5.5.1 5.5.1 5.1.2 5.1.3 5.5.2 5.5.3 5.5.3 6.1.1 5.1.3 6.2.2 6.2.6 2.5.3

Lernbereich 2: Algorithmik und Modellierung – prozedurale Programmierung (50 UE)

Die Schüler*innen lernen Programmieren als zentrale Methode des informatischen Denkens kennen – nicht als rein technische Fertigkeit, sondern als kreativen, strukturierten Zugang zur Problemlösung. Sie erkennen, dass Informationen in formalen Sprachen verarbeitet werden, und analysieren dabei den Aufbau natürlicher und künstlicher Sprachen. Auf dieser Basis modellieren sie reale Probleme und entwickeln daraus informatische Modelle, die als Bauplan für ihre Programme dienen. Beim Programmieren mit einer prozeduralen Sprache strukturieren sie Eingabe, Verarbeitung (Algorithmus) und Ausgabe. Sie nutzen visuelle Darstellungen wie Programmablaufpläne zur Planung und reflektieren auch die Benutzerschnittstellen (UI).

Durch eigenständiges Arbeiten mit unterstützenden Hilfesystemen fördern sie ihre Selbstlernkompetenz. Abschließend präsentieren und diskutieren sie ihre Lösungen, reflektieren gesellschaftliche Bezüge und entwickeln Fach-, Methoden- und Teamkompetenzen weiter.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Sprachverständnis und Sprachkonzepte - analysieren und vergleichen natürlicher und künstlicher Sprachen im Hinblick auf Syntax, Semantik, Universalität und Fehlerfreundlichkeit - untersuchen, wie Sprache über visuelle, auditive oder haptische Kanäle codiert und interpretiert werden kann (z. B. Gebärdensprache, Spracherkennung, Emojis) - erkennen, dass Programmiersprachen auf klar definierten Regeln (Grammatik, Syntax, Semantik) basieren - nutzen Begriffe wie Zeichen, Zeichenfolge, Alphabet, Variable, Methode und Parameter zur Beschreibung sprachlicher Strukturen in Programmen - reflektieren über die Bedeutung von Sprache für Kommunikation mit digitalen Systemen (z. B. Chatbots, Sprachassistenten, KI-Systeme) Problemlösen und Programmieren - analysieren komplexe Problemstellungen und zerlegen diese in logische Teilaufgaben - planen Lösungsideen mithilfe von Pseudocode, Struktogrammen oder Ablaufdiagrammen - entwickeln eigene Programme mit Kontrollstrukturen wie Sequenz, Verzweigung und Schleifen - setzen verschiedene Datentypen ein (Zahl, Text, Wahrheitswert, Gleitkommazahl) - schreiben lesbaren, klar strukturierten Quelltext mit sinnvollen Bezeichnern und Kommentaren - testen Programme systematisch, interpretieren Fehlermeldungen und verbessern den Code iterativ - nutzen moderne Entwicklungsumgebungen und Hilfesysteme zum selbstständigen Arbeiten Algorithmik und Modellierung - entwerfen Algorithmen in unterschiedlichen Darstellungsformen (verbal, grafisch, Quelltext)	Sprachverständnis und Sprachkonzepte - analysieren die Merkmale natürlicher und künstlicher Sprachen und vergleichen deren Aufbau und Einsatzmöglichkeiten - untersuchen alternative Codierungsformen von Sprache und interpretieren deren Bedeutung für barrierefreie Kommunikation - erklären Sprachkonzepte und wenden diese zur Beschreibung logischer Abläufe in Programmiersprachen an Problemlösen und Programmieren - analysieren Probleme und strukturieren diese in bearbeitbare Teilaufgaben - entwickeln Algorithmen und setzen sie mit geeigneten Programmiermitteln systematisch um - gestalten Quelltexte verständlich, nachvollziehbar und wartungsfreundlich - testen Programme gezielt und beheben Fehler eigenständig und reflektiert Algorithmik und Modellierung - entwickeln Algorithmen in geeigneter Form und begründen die Wahl der Darstellung - analysieren bestehende Programme und modifizieren diese zielgerichtet - übertragen reale Problemstellungen in algorithmische Modelle und bewerten deren Eignung	6.1.2 5.5.1 6.2.6 5.5.2 5.1.2 5.1.3 5.5.3 3.2.2 5.1.3 5.5.3 5.1.3 5.5.2

• achten dabei auf Ausführbarkeit, Eindeutigkeit und Terminierung • verstehen vorhandene Programme und passen sie an neue Anforderungen an • vergleichen verschiedene Lösungswege und optimieren diese • übertragen reale Probleme (z. B. Prozesse aus Alltag oder Umwelt) in informatische Modelle Einführung in objektnahe Konzepte (optional) • verwenden erste objektnahe Strukturen wie Attribute, Methoden und Parameter • erkennen die Vorteile modularer Programmierung (Wiederverwendbarkeit, klare Struktur, Zuständigkeitsverteilung) • beschreiben Objekte mit Zuständen (Attribute) und Verhalten (Methoden) • entwickeln einfache Programme mit Rückgabewerten und parametrisierten Funktionen	Einführung in objektnahe Konzepte (optional) • verwenden Attribute, Methoden und Parameter zur Strukturierung von Programmen • erklären die Grundprinzipien modularer Programmierung und wenden sie in einfachen Szenarien an • beschreiben Softwarebestandteile anhand ihrer Zustände und Funktionen	5.5.2 5.5.3 5.5.1
--	---	--

Lernbereich 3: Projekt – Softwareentwicklung oder Robotik (30 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
- Entwicklung eines eigenen kleinen Programms oder Spiels, das sich an Interessen aus dem Alltag orientiert (z. B. Rechenspiel, Quiz, Timer, Minigame) - alternativ: planen und umsetzen eines einfachen Robotik-Projekts mit selbst entwickelter Steuerung - Verwendung von einfachen Skizzen, Ablaufplänen oder Diagrammen zur Vorbereitung und Erklärung der Programmidee - programmieren mit vertrauten Bausteinen: Schleifen, Bedingungen, Variablen, Funktionen – und bei Bedarf: erste Datenlisten (Arrays) - testen, ob das Programm das ausführt, was es ausführen soll, Probleme erkennen und verbessern - präsentieren des fertigen Projekts vor der Klasse oder in kleiner Runde	- planen ein kleines digitales Projekt im Team und setzen eigene Ideen Schritt für Schritt um - strukturieren Aufgaben mithilfe einfacher Planungsmethoden und besprechen Lösungswege gemeinsam - programmieren grundlegende Abläufe mithilfe bekannter Strukturen wie Schleifen, Bedingungen und Variablen - testen Programme eigenständig und beheben einfache Fehler durch Ausprobieren und logisches Denken - dokumentieren den Projektverlauf und präsentieren ihre Ergebnisse adressatengerecht und anschaulich	3.1.2 2.3.1 5.1.3 2.3.2 5.5.3 5.1.3 3.2.2 2.4.2

Informatik in der Klassenstufe 9

Objektorient denken, Daten verstehen

In der 9. Schulstufe vertiefen die Schülerinnen ihre Programmierkenntnisse, indem sie erstmals mit der objektorientierten Modellierung arbeiten. Sie lernen, reale Dinge (wie Fahrzeuge oder Nutzerkonten) in Objekte zu übersetzen, die Eigenschaften (Attribute) und Fähigkeiten (Methoden) besitzen. Dabei setzen sie einfache Klassenhierarchien mit Vererbung um und verstehen, wie Methoden definiert, aufgerufen und angepasst werden. Ergänzt wird dies durch den Einsatz von Arrays zur Organisation größerer Datenmengen. Neben der Programmierung beschäftigen sie sich auch mit dem Umgang und der Auswertung von Daten. Sie lernen, Tabellen zu strukturieren, Formeln sinnvoll anzuwenden und Daten gezielt zu filtern oder zu analysieren – auch mithilfe einfacher KI-gestützter Tools, etwa zur automatischen Diagrammerstellung oder Formelhervorhebung. Ein weiterer Schwerpunkt liegt auf Datenbanken: Die Schülerinnen erstellen relationale Datenmodelle, führen Abfragen mit SQL durch und reflektieren gesellschaftliche Fragen im Umgang mit großen Datenmengen. In einem abschließenden Projekt planen und programmieren sie im Team eine Anwendung (z. B. ein Buchungs- oder Verwaltungssystem) mit Benutzerschnittstelle und Datenbankbindung, auch mit Bezug zu Webtechnologien wie HTML oder PHP.

Lernbereich 1: Algorithmik und Modellierung – Objektorientierte Modellierung und Programmierung (30 UE)

Die Schüler*innen lernen, reale Objekte in Klassen zu abstrahieren und grundlegende Prinzipien der objektorientierten Programmierung anzuwenden. Sie entwickeln einfache Programme mit Methoden, Attributen und Kontrollstrukturen, speichern Objekte in Listen und nutzen grafische Planungsmittel. Zusätzlich verstehen sie das Konzept der Vererbung zur Strukturierung und Wiederverwendung von Code und setzen unterstützende Werkzeuge der Softwareentwicklung ein.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Klassen modellieren und verstehen • entstehen, was Objekte, Klassen, Attribute und Methoden sind • analysieren reale Dinge (z. B. Fahrzeuge, Personen, Geräte) hinsichtlich Eigenschaften und Verhalten • abstrahieren Objekte in Klassen und stellen diese mithilfe von Klassendiagrammen dar • planen erste einfache objektorientierte Programme anhand alltagsnaher Beispiele • nutzen Variablen in verschiedenen Rollen: Attribut, Parameter, lokale Variable • erstellen und verwenden Konstruktoren zur Objekterzeugung • setzen Getter-/Setter-Methoden zur gezielten Zustandsänderung ein	Klassen modellieren und verstehen • analysieren reale Objekte aus ihrem Alltag hinsichtlich Eigenschaften (Attribute) und Verhalten (Methoden) • deklarieren Klassen in einer objektorientierten Programmiersprache mit passenden Attributen und Methoden • verwenden Variablen in unterschiedlichen Rollen (Attribut, Parameter, lokale Variable) zur Strukturierung von Programmen • erstellen Konstruktoren zur Objekterzeugung und erklären deren Bedeutung für den Objektzustand • ändern Attributwerte gezielt durch Wertzuweisungen oder Standardmethoden	5.5.1 5.5.3 5.5.2 5.5.1 5.5.3
Algorithmen objektorientiert entwickeln • formulieren einfache Methoden zur Beschreibung typischer Objektaktionen • unterscheiden zwischen Methodendefinition und Methodenaufruf • entwickeln Algorithmen mit bekannten Kontrollstrukturen (Sequenz, Verzweigung, Schleife) • verwenden grundlegende Datentypen (Zahl, Text, Wahrheitswert, Zeichenkette) zur Datenverarbeitung • planen Programmlogik grafisch (z. B. Pseudocode, Struktogramm, Sequenzdiagramm) • speichern und verarbeiten mehrere Objekte mit Arrays oder Listen	Algorithmen objektorientiert entwickeln • entwickeln einfache Algorithmen zur Beschreibung von Objektverhalten unter Verwendung von Kontrollstrukturen • formulieren und implementieren Methoden zur Ausführung typischer Aktionen eines Objekts • unterscheiden zwischen Methodendefinition und Methodenaufruf in objektorientierten Programmen • speichern mehrere Objekte in Arrays oder Listen und greifen gezielt auf deren Inhalte zu • planen und dokumentieren Programmabläufe mit grafischen Hilfsmitteln	5.5.3 5.5.3 5.5.2 5.5.3 5.1.3 3.2.2
Vererbung und Generalisierung • erkennen Gemeinsamkeiten und Unterschiede zwischen Klassen • strukturieren Programme durch Ober- und Unterklassen • nutzen Vererbung zur Weitergabe von Attributen und Methoden • erhalten Einblick in unterstützende Tools der Softwareentwicklung (z. B. Codevervollständigung, Syntaxprüfung durch KI)	Vererbung und Generalisierung • erkennen Gemeinsamkeiten und Unterschiede zwischen Klassen und strukturieren diese als Ober- und Unterklassen • modellieren einfache Vererbungsstrukturen mithilfe von Generalisierung und Spezialisierung • implementieren Klassenhierarchien unter Nutzung von Vererbung zur Wiederverwendung von Code • verwenden unterstützende digitale Werkzeuge (z. B. Codevervollständigung, Debugging-Tools, Versionskontrolle) zur Verbesserung der Entwicklungsprozesse	5.5.2 5.5.3 5.5.2 5.2.1 5.2.4

Lernbereich 2: Daten erfassen, analysieren und organisieren (30 UE)

Die Schüler*innen erwerben Kompetenzen im Umgang mit digitalen Tabellen und Datenbanksystemen. Sie lernen, Daten strukturiert zu erfassen, gezielt auszuwerten und mithilfe von Formeln, Filtern und logischen Funktionen zu analysieren. Dabei setzen sie auch erste automatisierte Werkzeuge ein. In der Datenmodellierung entwickeln sie einfache objektorientierte und relationale Strukturen, setzen relationale Datenbanken praktisch um und formulieren grundlegende SQL-Abfragen.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Tabellen und Datenanalyse-Tools • erstellen und strukturieren komplexerer Tabellen mit mehreren Datenreihen (z. B. in Excel, Google Sheets oder ähnlichen Plattformen) • anwenden von Sortier- und Filterfunktionen zur gezielten Datenauswertung (mehrstufig, mit mehreren Kriterien) • gestalten von Tabellen mithilfe sinnvoller Formatierungen (z. B. bedingte Formatierung bei Schwellenwerten) • automatisieren von Datenreihen (z. B. durch Drag-Funktionen oder KI-gestützte Vervollständigung) • analysieren von Daten mithilfe grundlegender Funktionen: min(), max(), mittelwert(), summe(), anzahl() • verwenden von logischen Funktionen wie wenn(), zählenwenn(), summewenn(), um einfache Auswertungslogik zu erstellen • erkennen und erklären den Unterschied zwischen absoluten und relativen Zellbezügen, z. B. beim Kopieren von Formeln • erhalten Einblick in aktuelle Werkzeuge zur automatisierten Datenanalyse (z. B. KI-gestützte Diagrammerstellung, Schnellanalysen) Datenmodellierung und Datenbanken • modellieren von einfachen Datenstrukturen anhand realer Szenarien (z. B. Buchungssystem, Inventar, Teilnehmerliste) • erstellen objektorientierter Datenmodelle mit Klassen, Attributen und Beziehungen (inkl. Kardinalität) • übertragen der Modelle in relationale Datenbankschemata mit Primär- und Fremdschlüsseln • implementieren einfacher Datenbanken in einem relationalen Datenbanksystem (z. B. SQLite, MySQL, Cloud-Dienste) • formulieren und testen grundlegender SQL-Abfragen (z. B. select, where, order by, join) zur gezielten Informationsgewinnung • erkennen und bewerten Datenprobleme wie Redundanz, Inkonsistenz oder Anomalien in fehlerhaft modellierten Datenbeständen • diskutieren Chancen und Risiken automatisierter Datenauswertung	Tabellen und Datenanalyse-Tools • planen und erstellen strukturierte digitale Tabellen zur Erfassung und Analyse von Daten • analysieren Daten mithilfe von Sortierung, Filtern, Formeln und einfachen logischen Funktionen • interpretieren Tabelleninhalte und ziehen daraus Schlüsse zur Verbesserung, Optimierung oder Visualisierung • erklären den Unterschied zwischen absoluten und relativen Zellbezügen und wenden diesen korrekt an Datenmodellierung und Datenbanken • modellieren Daten aus realen Szenarien in Form von objektorientierten und relationalen Strukturen • übertragen objektorientierte Datenmodelle in relationale Datenbankschemata unter Berücksichtigung von Schlüsseln und Beziehungen • implementieren einfache relationale Datenbanken und formulieren zielgerichtete SQL-Abfragen • bewerten Redundanz und Konsistenzprobleme in Datenbanken und reflektieren deren Auswirkungen • beurteilen den Einsatz datenbasierter Systeme unter ethischen und gesellschaftlichen Gesichtspunkten (z. B. Datenschutz, algorithmische Fairness, Transparenz)	1.3.2 3.1.2 1.2.1 5.5.2 6.1.1 6.2.2 5.5.2 5.5.2 5.5.3 5.2.1 5.5.3 6.2.2 6.2.5 6.2.6 4.2.1

Lernbereich 3: Projekt – Softwareentwicklung oder Datenbank (30 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
- entwickeln eines Softwareprojekts zu einem praxisnahen Szenario (z. B. Buchungs-, Inventar-, Kontakt- oder Verwaltungssystem) - planen des Projektablaufs im Team: Arbeitsteilung, Meilensteine, Zeitrahmenmodellieren der Datenstrukturen als Klassendiagramm, ggf. ergänzt durch weitere Darstellungen (z. B. Benutzeroberfläche, Ablaufdiagramm) - programmieren eines objektorientierten Systems mit Datenhaltung – z. B. Speicherung von Benutzerdaten, Reservierungen oder Produkten - verwenden einer relationalen Datenbank (z. B. SQLite, MySQL) zur strukturierten Speicherung und Abfrage von Daten - formulieren und integrieren einfacher SQL-Abfragen (z. B. Anzeige, Filterung, Sortierung von Daten) - gestalten einer grafischen Benutzeroberfläche oder – in der Vertiefung – Anbindung über einfache Webtechnologien (z. B. HTML/PHP) - testen das Programm auf Funktionalität, Korrektheit und Benutzerfreundlichkeit - dokumentieren die Projektentwicklung mit Screenshots, Ablaufplänen, kurzen Texten oder Präsentationen	- planen ein Softwareentwicklungsprojekt im Team und legen dafür sinnvolle Arbeitsschritte und Zuständigkeiten fest - analysieren ein praxisnahes Szenario und strukturieren es mithilfe geeigneter Modellierungsmethoden (z. B. Klassendiagramm, UI-Skizze) - implementieren ein objektorientiertes Programm zur Lösung eines konkreten Anwendungsproblems - integrieren eine relationale Datenbank zur dauerhaften Datenspeicherung und verarbeiten Daten mithilfe einfacher SQL-Abfragen - gestalten ein einfaches, benutzerfreundliches Interface – optional auch als Weboberfläche - testen und optimieren die Anwendung anhand der Projektziele und dokumentieren Entwicklung und Ergebnisse adressatengerecht - reflektieren über den Softwareentwicklungsprozess und die Bedeutung von Teamarbeit, Struktur und Usability in realen IT-Projekten	2.3.1 3.1.2 5.1.3 5.5.3 3.2.2 5.5.3 5.2.1 5.5.2 3.1.2 6.1.1 5.1.3 3.2.2 2.4.2 6.2.2 2.3.2 6.2.6

Informatik in der Klassenstufe 10

Digitale Systeme entwerfen, programmieren und reflektieren

In der 10. Schulstufe vertiefen die Schüler*innen ihr Verständnis für objektorientierte Programmierung, strukturierte Datenverarbeitung und den Aufbau digitaler Anwendungen. Sie lernen, Daten mit Hilfe von Arrays oder Listen effizient zu speichern, zu verwalten und gezielt zu bearbeiten. Objektbeziehungen werden modelliert und mithilfe von Referenzen sowie Methodenaufrufen umgesetzt. Dabei erwerben sie grundlegende Kenntnisse zur Kapselung, Vererbung und zum Einsatz von Polymorphismus in flexiblen Softwarestrukturen. Im Bereich eingebetteter Systeme arbeiten sie praktisch mit Sensoren, analysieren deren Daten und entwickeln einfache Anwendungen, die analoge Signale digital verarbeiten. Sie wenden strukturiertes Vorgehen bei Planung, Dokumentation und Test an und reflektieren dabei auch Aspekte wie Sicherheit, Wartbarkeit und gesellschaftliche Verantwortung. In einem abschließenden Softwareprojekt setzen sie ihr Wissen anwendungsorientiert um.

Lernbereich 1: Strukturierte Datenverarbeitung und objektorientierte Programmierung (40 UE)

In diesem Lernbereich erwerben die Schülerinnen grundlegende Kenntnisse zur strukturierten Verwaltung von Daten und zur Gestaltung objektorientierter Programme. Sie nutzen Datenstrukturen wie Arrays oder Listen, um gleichartige Objekte effizient zu speichern und zu bearbeiten. Dabei modellieren sie Objektbeziehungen mit unterschiedlichen Kardinalitäten und setzen diese mithilfe von Referenzen und Methodenaufrufen um. Der kontrollierte Zugriff auf Attribute durch Datenkapselung wird ebenso behandelt wie Prinzipien guter Softwarepraxis. Darüber hinaus vertiefen die Schülerinnen ihr Verständnis für Vererbung und Polymorphismus, entwickeln flexible Klassenhierarchien und nutzen polymorphe Aufrufe zur dynamischen Steuerung von Verhalten. So entstehen Grundlagen für die Entwicklung modularer und wartbarer Softwarelösungen.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Daten verwalten mit Arrays und Objekten • verwenden eindimensionale indizierte Datenstrukturen (z. B. Arrays, Listen) zur Speicherung gleichartiger Datenobjekte • verstehen Begriffe wie Index, Element, Länge und deren Rolle bei der Verwaltung von Datenreihen • modellieren Verwaltungsstrukturen für Objekte (z. B. Mitgliedslisten, Buchungseinträge, Artikellisten) • entwickeln Methoden zum Einfügen, Entfernen und Bearbeiten von Elementen in einer strukturierten Sammlung • diskutieren Vor- und Nachteile fester und dynamischer Datenstrukturen im Vergleich zu moderner Speicherlogik (z. B. Listen, Datenbanken) Objektbeziehungen, Referenzen und kontrollierter Zugriff • analysieren und modellieren Beziehungen zwischen Objekten mit unterschiedlichen Kardinalitäten (z. B. 1:1, 1:n) • setzen Klassenbeziehungen mithilfe von Referenzen um (z. B. ein Fahrradverleih mit Kunden und Fahrrädern) • ermöglichen Interaktionen zwischen Objekten durch Methodenaufrufe (z. B. Reservierung, Rückgabe, Anzeige) • erkennen die Notwendigkeit von Kapselung und kontrolliertem Zugriff auf Attribute (z. B. durch Getter/Setter oder Sichtbarkeiten) • reflektieren über gute Praxis in der Softwareentwicklung (z. B. Schutz von Daten, Entkopplung, einfache Wartung) Vererbung, Polymorphismus • erweitern bestehende Klassenmodelle durch Vererbung in Generalisierungshierarchien • nutzen Polymorphismus zur Implementierung unterschiedlicher Verhaltensweisen in Unterklassen (z. B. Berechnung von Beiträgen je nach Mitgliedstyp)	Daten verwalten mit Arrays und Objekten • wenden eindimensionale Datenstrukturen zur Speicherung und Verwaltung gleichartiger Objekte an • modellieren objektorientierte Verwaltungsstrukturen anhand konkreter Beispiele aus dem Alltag Objektbeziehungen, Referenzen und kontrollierter Zugang • analysieren und setzen Beziehungen zwischen Objekten mithilfe von Referenzen und passenden Methoden um • erklären die Bedeutung von Datenkapselung und wenden kontrollierten Zugriff auf Attribute korrekt an • entwickeln Methoden zur Bearbeitung von Datenobjekten (Einfügen, Löschen, Ändern) in einer Objektliste Vererbung, Polymorphismus • wenden Vererbung und Polymorphismus zur Umsetzung differenzierter Verhaltensweisen innerhalb einer Klassenhierarchie an • implementieren polymorphe Methoden und erklären deren Einsatz zur flexiblen Steuerung von Objektverhalten • reflektieren den praktischen Nutzen dieser Konzepte für wiederverwendbaren, wartbaren und strukturierten Code in realen Anwendungsszenarien	 5.5.2 5.5.3 5.5.2 5.5.3 5.5.1 5.5.2 5.5.3 5.5.2 5.5.3 6.2.2 5.1.3

• überschreiben Methoden gezielt und nutzen gemeinsame Schnittstellen für unterschiedliche Objekttypen • setzen polymorphe Methodenaufrufe zur Laufzeit situationsgerecht ein und verstehen deren Bedeutung für flexible Programmstrukturen • erkennen die Bedeutung dieser Konzepte für die Entwicklung skalierbarer, modularer Anwendungen (z. B. Online-Shops, Benutzerverwaltung)		
---	--	--

Lernbereich 2: Sensorbasierte Systeme (20 UE)

Zahlreiche digitale Systeme im Alltag verarbeiten Informationen, die sie über Sensoren erfassen. Durch die praktische Arbeit mit solchen Sensoren verstehen die Schüler*innen, wie analoge Signale in digitale Daten umgewandelt werden, ein zentrales Prinzip der Digitalisierung. Dabei vertiefen sie ihre algorithmischen Kompetenzen mithilfe einer blockbasierten Programmiersprache.

Ein typisches Anwendungsszenario ist das Programmieren eines Mikrocontrollers oder eines autonomen Systems, das Messdaten aufnimmt und verarbeitet. In diesem Zusammenhang setzen die Schüler*innen den Datentyp *Liste* ein, um Werte strukturiert zu speichern und auszuwerten. Das EVAS-Modell dient dabei als vereinfachtes, abstrahiertes Denkmodell zur Beschreibung grundlegender Abläufe in einem digitalen System.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Sensorik und Datenaufnahme in digitalen Systemen • erkennen und benennen von Sensoren in digitalen Alltagsgeräten (z. B. Temperatur-, Bewegungs-, Lichtsensoren in Smartphones, Wearables, Smart Homes) • bestimmen des Wertebereichs von Sensoren und interpretieren von Rohdaten im Kontext der gemessenen Größe • erklären des Prinzips der Analog-Digital-Wandlung anhand praktischer Beispiele (z. B. Temperaturmessung in °C in digitale Zahl) • diskutieren die Bedeutung von Messintervallen, Abstraten und Genauigkeit im Rahmen datenbasierter Entscheidungen Datenverarbeitung mit strukturierten Methoden • abfragen, speichern und verarbeiten von Sensorwerten in Listen oder Arrays • anwenden grundlegender Operationen wie Hinzufügen, Löschen, Suchen und Durchschnitte bilden • analysieren und visualisieren von Messdaten im Programm oder mithilfe externer Tools (z. B. Tabellenkalkulation, Diagrammerstellung) • erkennen von Datenfehlern und Schwankungen sowie Diskussion über Zuverlässigkeit und Kalibrierung von Sensorwerten Softwareentwicklung für eingebettete Systeme • verstehen und prüfen von Spezifikationen als Grundlage für eine zielgerichtete Programmumsetzung • entwickeln einfacher Anwendungen für Mikrocontroller oder vergleichbare Systeme (z. B. Messung, Alarm, Steuerung) • dokumentieren die Softwarelösung durch Kommentare im Code, kurze Projektberichte oder digitale Präsentationen • testen die Anwendung systematisch und leiten aus den Testergebnissen Verbesserungen und neue Anforderungen ab	Sensorik und Datenaufnahme in digitalen Systemen • identifizieren Sensoren in digitalen Systemen und beschreiben deren Funktion sowie den zugehörigen Wertebereich • erklären die Umwandlung kontinuierlicher Messgrößen in digitale Daten anhand einfacher Analog-Digital-Beispiele Datenverarbeitung mit strukturierten Methoden • verarbeiten Sensorwerte mit geeigneten Datenstrukturen (z. B. Listen) zur systematischen Speicherung und Analyse • analysieren Messdaten auf Auffälligkeiten, Genauigkeit und Zuverlässigkeit und beurteilen ihre Aussagekraft Softwareentwicklung für eingebettete Systeme • planen einfache Anwendungen basierend auf einer Spezifikation und erweitern diese bei Bedarf selbstständig • dokumentieren Softwarelösungen nachvollziehbar durch Kommentare, Ablaufbeschreibungen oder Nutzerhinweise • testen entwickelte Programme strukturiert und reflektieren die Ergebnisse zur Optimierung der Funktionalität • leiten aus ihren Projekten Rückschlüsse auf Sicherheit, Fehlertoleranz und nachhaltige Konstruktion digitaler Systeme ab • diskutieren gesellschaftliche Aspekte datenbasierter Technologien (z. B. Datenschutz, Datenmanipulation, Vertrauen in Sensorik)	 5.5.1 5.5.1 5.5.2 1.3.2 1.2.1 6.1.3 5.1.3 5.5.3 3.2.2 5.1.3 4.1.1 6.2.2 6.2.5 6.2.6 4.2.1

Lernbereich 3: Künstliche Intelligenz (10 UE)

Die Schüler*innen setzen sich mit den Grundlagen Künstlicher Intelligenz auseinander. Sie lernen Aufbau und Funktionsweise neuronaler Netze kennen, vergleichen verschiedene Lernverfahren wie überwachtes, unüberwachtes und bestärkendes Lernen und wenden diese in einfachen Anwendungen an. Mithilfe schulgeeigneter Tools simulieren sie Lernprozesse, analysieren den Einfluss von Trainingsdaten und Parametern und reflektieren kritisch die Qualität und Aussagekraft von KI-Ergebnissen. Abschließend diskutieren sie gesellschaftliche und ethische Fragestellungen wie Fairness, Verantwortung und mögliche Risiken im Umgang mit KI.

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
Neuronale Netze - Aufbau neuronaler Netze: Schichten, Gewichtungen, Aktivierungsfunktionen - Vorwärtsausbreitung und Backpropagation - Einfluss von Netzarchitektur und Parametern - Visualisierung und Simulation in Tools wie VibeCoding oder Teachable Machine Lernverfahren im Vergleich - Überwachtes, unüberwachtes und bestärkendes Lernen - Beispiele: Klassifikation, k-Means, einfache Agentensimulationen - Einfluss von Trainingsdaten, Hyperparametern, Bias - Umsetzung mit geeigneten KI-Werkzeugen Gesellschaftliche Dimension - Bewertung von Ergebnissen: Validität, Overfitting, Robustheit - Reflexion über Risiken: Diskriminierung, Deepfakes, Kontrollverlust - Diskussion zu Ethik, Fairness und Verantwortung in KI-Systemen	Neuronale Netze - erklären Aufbau und Funktionsweise neuronaler Netze und simulieren Lernprozesse. Lernverfahren im Vergleich - vergleichen zentrale Lernverfahren und setzen einfache Beispiele praktisch um. - analysieren Einflussfaktoren auf Lernerfolg und interpretieren Ergebnisse kritisch. - nutzen schulgeeignete KI-Tools zur Umsetzung konkreter Anwendungen. Gesellschaftliche Dimension - reflektieren gesellschaftliche und ethische Aspekte des KI-Einsatzes.	5.5.1 5.5.3 5.5.2 5.2.1 1.2.1 6.2.2 5.2.1 5.2.4 6.2.5 6.2.6

Lernbereich 3: Projekt IV – digitale Anwendung entwickeln (20 UE)

Inhalt	Kompetenzen: Die Schüler*innen ...	KMK
• konzipieren und realisieren eines digitalen Projekts im Team zu einem praxisnahen Thema aus der Lebenswelt der Schüler*innen (z. B. Buchungssystem, Info-Anwendung, interaktives Quiz, Verwaltungs- oder Spielsystem) • planen des Projektablaufs: Zieldefinition, Arbeitsteilung, Zeitmanagement und Dokumentation • analysieren und modellieren der Aufgabenstellung mit geeigneten Methoden (z. B. Klassendiagramme, Skizzen, Ablauf- oder Zustandsdiagramme) • entwickeln einer funktionalen Anwendung mithilfe objektorientierter Programmierung (z. B. Klassen, Methoden, Vererbung, einfache Benutzeroberfläche) • verarbeiten und verwalten von strukturierten Daten in Listen, Objektsammlungen oder einfachen Dateien • gestalten benutzerfreundlicher Interfaces – grafisch oder webbasiert (z. B. mit HTML/CSS/PHP oder blockbasierten UI-Tools) • testen die Anwendung systematisch anhand von Beispieldaten und vordefinierten Anforderungen • dokumentieren den Entwicklungsprozess und das Endprodukt in geeigneter Form (z. B. Präsentation, Screencast, schriftliche Anleitung) • reflektieren über Benutzerfreundlichkeit, Wartbarkeit, Fehleranfälligkeit und Qualität der Lösung • diskutieren gesellschaftlich relevante Aspekte digitaler Systeme (z. B. Barrierefreiheit, Datenschutz, Informationsverantwortung)	• planen ein Softwareprojekt zielgerichtet im Team und übernehmen Verantwortung für Teilaufgaben • analysieren ein Szenario und modellieren zentrale Bestandteile der Anwendung mit passenden Darstellungen • entwickeln eine objektorientierte Anwendung mit klaren Strukturen und benutzerfreundlicher Oberfläche • verwenden geeignete Datenstrukturen zur Organisation von Informationen (z. B. Listen, Arrays, Dateien) • testen die entwickelte Software lösungsorientiert und ziehen Rückschlüsse auf Funktionsweise und mögliche Fehlerquellen • dokumentieren die Projektergebnisse adressatengerecht und reflektieren den Entwicklungsprozess kritisch • beurteilen die technische und gesellschaftliche Qualität digitaler Anwendungen in Bezug auf Funktionalität, Nutzbarkeit und Verantwortung	2.3.1 5.1.3 3.1.2 5.5.3 3.2.2 5.5.3 3.1.2 5.5.2 5.1.3 3.2.2 6.2.2 6.2.2 6.2.2 6.2.6

Wahlfach Informatik in den Klassenstufen 9 & 10

Das Wahlfach Informatik in den Jahrgangsstufen 9 und 10 richtet sich an Schülerinnen und Schüler der DSM, die in Klasse 7 Französisch gewählt und daher nicht am Wahlpflichtfach Informatik der Jahrgänge 8-10 teilgenommen haben. Ziel ist es, bis zum Ende der 10. Klasse den gleichen fachlichen Stand zu erreichen wie die Lernenden des Wahlpflichtfachs, sodass ein nahtloser Übergang in die weiterführenden Informatikmodule der Oberstufe (Jahrgangsstufen 11 und 12) möglich ist.

Dazu behandelt das Wahlfach verpflichtend vier zentrale Themenfelder, die sich an den Inhalten und Kompetenzerwartungen des regulären Informatikunterrichts der Klassen 8 bis 10 orientieren:

- **Lernbereich 1: Algorithmik und Modellierung mit prozeduralen Sprachen**
Einführung in algorithmisches Denken, Strukturierung von Programmen mit Kontrollstrukturen, Variablen, Funktionen und ersten grafischen Darstellungen (z. B. Struktogramme, Pseudocode).
- **Lernbereich 2: Grundlagen der objektorientierten Modellierung und Programmierung**
Aufbau von Klassen, Attributen, Methoden und objektorientierten Strukturen; Beziehungen zwischen Objekten, Datenkapselung sowie einfache Vererbung und Polymorphismus.
- **Lernbereich 3: Datenverarbeitung, Datenanalyse und Tabellenkalkulation**
Die Inhalte bereiten auf datenbezogene Fragestellungen der Oberstufe und auf gesellschaftliche Themen wie Dateninterpretation, Automatisierung und Entscheidungsunterstützung vor.
- **Lernbereich 4: Datenstrukturen und einfache Datenspeicherung**
Speicherung und Auswertung strukturierter Daten (z. B. mit Arrays, Listen), Einführung in Datenmodellierung, tabellarische Datenverarbeitung und grundlegende Prinzipien relationaler Datenhaltung.

Der Unterricht ist projektorientiert und schülerzentriert aufgebaut. Unterschiedliche Lernniveaus werden durch gezielte Differenzierung berücksichtigt. Leistungsstarke Schülerinnen und Schüler können zusätzliche Aufgaben oder Mini-Projekte übernehmen, um gezielt Kompetenzen im Bereich der Softwareentwicklung und digitalen Systemgestaltung zu vertiefen. Auf diese Weise wird eine solide Grundlage geschaffen, um spätere Inhalte der Qualifikationsphase fachlich sicher zu bewältigen.

5. Bewertungskriterien und Überprüfbarkeit von Lernleistungen

Die Fachkonferenz legt folgende Grundsätze für die Bewertung im Fach Informatik fest:

- Verhältnis von schriftlichen und mündlichen Leistungen:
Es wird festgelegt, wie schriftliche (z. B. Klassenarbeiten) und mündliche Leistungen (z. B. Beiträge im Unterricht, Präsentationen) gewichtet werden sollen, um die Gesamtnote zu ermitteln.
- Transparente Bewertung von Klassenarbeiten:
Es werden prozentuale Vorgaben oder konkrete Punktabgaben für die Notenvergabe in Klassenarbeiten gemacht.
- Weitere Aspekte der Bewertung:
Es wird vereinbart, wie sprachliche Ausdrucksfähigkeit in die Bewertung einfließt oder in welchem Umfang grundlegendes Fachwissen abgefragt werden soll.

5.1. Anpassung an Schulformen:

Bei der Gestaltung und Bewertung von Klassenarbeiten sind die Schulart und das entsprechende Anforderungsniveau zu berücksichtigen:

- Realschule: Der Anforderungsbereich III (komplexe Problemstellungen, Bewertung von Sachverhalten) wird nur in geringem Umfang geprüft.
- Hauptschule: Der Fokus liegt verstärkt auf dem Anforderungsbereich I (Grundwissen und einfache Anwendung).
- Die Bewertung der mündlichen Mitarbeit wird entsprechend angepasst.

5.2. Definition der Anforderungsbereiche:

- Anforderungsbereich I: Reproduktion
Wiedergabe einfacher Fachinhalte, Anwendung grundlegender Methoden, Darstellung in vorgegebener Form
- Anforderungsbereich II: Reorganisation und Transfer
Umstrukturierung und Anwendung komplexerer Inhalte und Methoden, situationsgerechte Kommunikation, einfache Bezüge und Bewertungen.
- Anforderungsbereich III: Problemlösen und Bewerten
Selbstständige Anwendung komplexer Inhalte und Methoden, Auswahl passender Darstellungsformen, begründete Bewertungen und Verknüpfungen von Sachverhalten

5.3. Leistungsbewertung

Jahr- gang	Art der Leistungsbewertung	Gewichtungen
5	Es werden ausschließlich mündliche Leistungen erbracht (z. B. Unterrichtsbeiträge, praktische Arbeiten am Rechner, Tests).	Gewichtung der einzelnen Bestandteile liegt im Ermessen der Lehrkraft.
6	Es werden ausschließlich mündliche Leistungen erbracht (z. B. Unterrichtsbeiträge, praktische Arbeiten am Rechner, Tests).	Gewichtung der einzelnen Bestandteile liegt im Ermessen der Lehrkraft.
7	Es werden ausschließlich mündliche Leistungen erbracht (z. B. Unterrichtsbeiträge, praktische Arbeiten am Rechner, Tests).	Gewichtung der einzelnen Bestandteile liegt im Ermessen der Lehrkraft.
8	Wahlpflichtfach: - Schriftliche Leistungen: 1 Klassenarbeit pro Halbjahr (Projekt kann Klassenarbeit ersetzen) - Mündliche Leistungen: Unterrichtsbeiträge, Präsentationen, Tests, praktische Arbeiten am Rechner	Wahlpflichtfach: 1:1
9	Wahlpflichtfach: - Schriftliche Leistungen: 1 Klassenarbeit pro Halbjahr (Projekt kann Klassenarbeit ersetzen) - Mündliche Leistungen: Unterrichtsbeiträge, Präsentationen, Tests, praktische Arbeiten am Rechner Wahlfach: Es werden nur mündliche Leistungen in Form von Unterrichtsbeiträgen, praktischen Arbeiten am Rechner, angekündigten und unangekündigten Tests etc. erhoben.	Wahlpflichtfach: 1:1 Wahlfach: Gewichtung der einzelnen Bestandteile liegt im Ermessen der Lehrkraft.
10	Wahlpflichtfach: - Schriftliche Leistungen: 1 Klassenarbeit pro Halbjahr (Projekt kann Klassenarbeit ersetzen) - Mündliche Leistungen: Unterrichtsbeiträge, Präsentationen, Tests, praktische Arbeiten am Rechner Wahlfach: Es werden nur mündliche Leistungen in Form von Unterrichtsbeiträgen, praktischen Arbeiten am Rechner, angekündigten und unangekündigten Tests etc. erhoben.	Wahlpflichtfach: 1:1 Wahlfach: Über die einzelne Gewichtung entscheidet die Lehrkraft individuell.