

Bildungsplan 2016

Bildungspläne 2016

Gymnasium

Schulversuchsfassung

Informatik (Schulversuch)

Stand: 21. Januar 2020

Stuttgart 2023

Impressum

Herausgeber: Ministerium für Kultus, Jugend und Sport Baden-Württemberg,
Postfach 103442, 70029 Stuttgart

Urheberrecht: Die fotomechanische oder anderweitig technisch mögliche Reproduktion des Satzes beziehungsweise der Satzordnung für kommerzielle Zwecke bedarf der Genehmigung des Herausgebers.

Inhaltsverzeichnis

1. Leitgedanken zum Kompetenzerwerb	5
1.1 Bildungswert des Faches	5
1.2 Kompetenzen	7
1.3 Didaktische Hinweise	9
1.4 Wege in die Kursstufe über Profulfach IMP oder Brückenkurs	11
2. Prozessbezogene Kompetenzen	12
2.1 Strukturieren und Vernetzen	12
2.2 Modellieren und Implementieren	13
2.3 Kommunizieren und Kooperieren	14
2.4 Analysieren und Bewerten	15
3. Standards für inhaltsbezogene Kompetenzen	16
3.1 Klassen 9/10 (Brückenkurs)	16
3.1.1 Daten und Codierung	16
3.1.2 Algorithmen	17
3.1.3 Rechner und Netze	19
3.1.4 Informationsgesellschaft und Datensicherheit	20
3.2 Klassen 11/12 (Basisfach)	22
3.2.1 Daten und Codierung	22
3.2.1.1 Bitebene	22
3.2.1.2 Datenstrukturen	23
3.2.1.3 Relationale Datenbanksysteme	24
3.2.2 Algorithmen	25
3.2.2.1 Strukturierte und objektorientierte Programmierung	25
3.2.2.2 Algorithmen auf Datenstrukturen	27
3.2.2.3 Rekursion	28
3.2.3 Rechner und Netze	29
3.2.4 Informationsgesellschaft und Datensicherheit	30
3.2.5 Automaten und formale Sprachen	31
3.2.6 Projektmanagement	32
3.3 Klassen 11/12 (Leistungsfach)	33

3.3.1 Daten und Codierung	33
3.3.1.1 Bitebene	33
3.3.1.2 Datenstrukturen	34
3.3.1.3 Relationale Datenbanksysteme	35
3.3.2 Algorithmen	36
3.3.2.1 Strukturierte und objektorientierte Programmierung	36
3.3.2.2 Algorithmen auf Datenstrukturen	38
3.3.2.3 Rekursion	40
3.3.3 Rechner und Netze	41
3.3.4 Informationsgesellschaft und Datensicherheit	43
3.3.5 Automaten und formale Sprachen	45
3.3.6 Projektmanagement	47
4. Operatoren	48
5. Anhang	50
5.1 Verweise	50
5.2 Abkürzungen	52
5.3 Geschlechtergerechte Sprache	55
5.4 Besondere Schriftauszeichnungen	55

1. Leitgedanken zum Kompetenzerwerb

1.1 Bildungswert des Faches

Informatik ist eine Wissenschaft, die strukturwissenschaftliche und ingenieurwissenschaftliche Komponenten verbindet. Sie beschäftigt sich dazu systematisch mit Konzepten zur Darstellung, Verarbeitung, Strukturierung und zum Transport von Informationen und nutzt diese Konzepte für die Entwicklung von Informatiksystemen. Die Informatik stellt heute einen organischen Teil vieler anderer Disziplinen dar und hat diese in kurzer Zeit verändert. Alltägliches Handeln wird ebenso von diesen Informatiksystemen gesteuert wie die lebensnotwendige Grundversorgung in den Bereichen Energie, Logistik, Transport und Kommunikation. Kinder und Jugendliche bewegen sich also in einer zunehmend digitalisierten Welt. Durch die Digitalisierung ist eine weitere Dimension der realen Welt und des Zusammenlebens entstanden. Einerseits haben viele nur durch die Informatik ermöglichten Anwendungen (wie zum Beispiel Navigationssysteme, Wissensdatenbanken, Kommunikationsplattformen, Unterhaltungselektronik, Streamingdienste, Onlineshopping und Cloud-Computing) unser Leben bereichert und vereinfacht. Andererseits birgt es auch Gefahren, wenn die automatisierte und algorithmengesteuerte Verarbeitung von Informationen durch massenhaftes Erheben und Verknüpfen von Daten bereits so in den Alltag integriert ist, dass eine Beeinflussung durch deren Prognosen und Handlungsempfehlungen häufig nicht mehr wahrgenommen wird. Ziel des Informatikunterrichts ist es, dass Schülerinnen und Schüler ein Verständnis für Hintergründe, Mechanismen und Funktionsweisen von informatischen Systemen entwickeln. Dabei ist es von großer Bedeutung, nicht nur zu wissen, wie Anwendungen genutzt werden, sondern auch ihre Funktionsweise zu verstehen. Bei der Erstellung von informatischen Produkten erleben die Schülerinnen und Schüler, wie sie selbst gestalterisch tätig werden können und erfahren ihre Selbstwirksamkeit. Ein Bewusstsein für die Existenz und Relevanz der Beeinflussungen durch informatische Systeme sowie die Erfahrung, informatische Systeme selbst mitgestalten zu können, tragen dazu bei, dass sie als mündige Bürgerinnen und Bürger in der Gesellschaft verantwortungsvoll Entscheidungen treffen können.

Beitrag des Faches zu den Leitperspektiven

In welcher Weise das Fach Informatik einen Beitrag zu den Leitperspektiven leistet, wird im Folgenden dargestellt:

- **Bildung für nachhaltige Entwicklung (BNE)**

Die Schülerinnen und Schüler setzen moderne Informations- und Kommunikationstechnologien ein und nehmen im Informatikunterricht auch zunehmend die gestalterische Rolle ein. Im Informatikunterricht werden Schülerinnen und Schüler mit der stets vorhandenen Begrenztheit von Ressourcen konfrontiert. Sie erfahren von Grund auf die Notwendigkeit, Methoden zu entwickeln und zu optimieren, um Ressourcen effizient einzusetzen. Viele im Alltag und auch im Informatikunterricht eingesetzte elektronische Geräte bieten Gesprächsanlässe, um Aspekte nachhaltiger Nutzung aus informatischer Perspektive zu diskutieren. Die Schülerinnen und Schüler entwickeln zunehmend ein Bewusstsein für nachhaltigkeitsfördernde Konzepte wie beispielsweise langfristig wartbare Systeme, komponentenweise austauschbare Systeme oder das Teilen von Ressourcen (zum Beispiel mittels Virtualisierung). Darauf aufbauend werden sie in der Kursstufe befähigt, sich zu aktuellen Fragestellungen im Hinblick auf tatsächliche oder vermeintliche Nachhaltigkeit zu positionieren. Aufgrund der zunehmenden Knappheit von Ressourcen auf der Erde wird deren effiziente Nutzung und der Rückgang des globalen Ressourcenverbrauchs eine der größten Herausforderungen der Zukunft sein. Um diese zu bewältigen, werden in vielen Bereichen grundlegend neue Ansätze erforderlich sein, bei denen informatische Konzepte eine zentrale Rolle spielen.

- **Bildung für Toleranz und Akzeptanz von Vielfalt (BTV)**

Informatiksysteme wirken in zahlreichen gesellschaftlichen Bereichen, zum Beispiel in der Fahrzeugtechnik, in der Medizintechnik, in der Kommunikationstechnik und im Zahlungsverkehr. Zunehmend fällen diese Systeme autonom Entscheidungen, die sich nicht oder nicht unmittelbar nachvollziehbar beeinflussen lassen, jedoch konkrete Auswirkungen auf Menschen haben. Die Entwickler derartiger Informatiksysteme tragen damit eine erhebliche gesellschaftliche Verantwortung. Schülerinnen und Schüler entwickeln im Informatikunterricht anhand konkreter Fragestellungen ein Verantwortungsbewusstsein für die vielfältigen Bedürfnisse verschiedener Gruppen der Gesellschaft: Wie können Algorithmen zur Bonitätseinstufung Personen diskriminieren? Welche ethischen Überlegungen müssen kollisionsvermeidenden Algorithmen für selbstfahrende Automobile zu Grunde liegen? Wie können automatische und manuelle Zensursysteme das Recht auf freie Meinungsäußerung tangieren? Die deutsche Bundesregierung hat mit ihrer Entscheidung, den „contract for the web“ zu unterzeichnen, deutlich gemacht, dass der Zugriff aufs Internet ein Grundrecht ist. Allerdings werden durch Ausübung dieses Grundrechts auch andere Grundrechte tangiert. Denn das Internet ist aufgrund seiner Topologie ein schwer zu kontrollierender Raum, in dem in der Folge Beleidigungen, Cybermobbing, Hetze und Intoleranz ihren Platz finden. Bestimmte Interessengruppen nutzen das Internet auch, um Meinungen und Ansichten zu verstärken, zu beeinflussen oder zu manipulieren. Ein gezielter Einfluss auf die öffentliche Meinung ist eine Gefahr für die Akzeptanz von gesellschaftlicher Vielfalt und fördert Vorurteile und Klischees. Nur wenn die Schülerinnen und Schüler die Strukturen des Internets und die dahinter stehenden technischen Möglichkeiten verstehen, können sie Informationen, die im Internet verfügbar sind, hinsichtlich ihres Wahrheitsgehalts und ihrer Vorurteilsfreiheit bewerten.

Agiles, kollaboratives Arbeiten in Projekten verlangt und übt den wertschätzenden Umgang miteinander und fördert gegenseitigen Respekt und Achtung von Vielfalt.

- **Prävention und Gesundheitsförderung (PG)**

Die Schülerinnen und Schüler der Kursstufe erlernen Strategien zur Lösung zunehmend komplexerer Problemstellungen – sowohl algorithmische Problemlösekompetenzen als auch Teamfähigkeit sind dabei gefordert. Bei der konkreten Realisierung von Lösungen müssen Konflikte bewältigt, Kompromisse gefunden und Ergebnisse vertreten werden. Bei der intensiven Auseinandersetzung mit Mitschülerinnen und Mitschülern an einer konkreten Aufgabe wird zur Förderung der Resilienz Selbstwirksamkeit erlebbar und Kompromissbereitschaft geübt. Rechner und mobile Endgeräte sind wichtige Werkzeuge im Informatikunterricht. Neben der reinen inhaltlichen Arbeit mit den Geräten ist die Arbeitshygiene ein weiterer wichtiger Aspekt bei ihrem unterrichtlichen Einsatz. In der Sekundarstufe I lernen die Schülerinnen und Schüler Strategien zur Gesunderhaltung, welche in der Kursstufe aufgegriffen werden, da Präventionsmaßnahmen nur bei regelmäßiger Anwendung oder Wiederholung ihre volle Wirksamkeit entfalten – insbesondere im Hinblick auf Suchtverhalten. Vorrangiges Ziel muss sein, dass die erlernten Strategien von den Schülerinnen und Schülern in den Alltag integriert werden. Dies gelingt umso mehr, wenn neben den positiven Effekten gesundheitsfördernder Maßnahmen in der Kursstufe zusätzlich die negativen Konsequenzen bei Missachtung dieser Maßnahmen aufgezeigt werden.

- **Berufliche Orientierung (BO)**

Mit der rasanten Entwicklung der Informationstechnologien geht eine ebenso rasante Entwicklung der zugehörigen und davon abhängigen Berufsfelder einher. Besuche von Betrieben und Hochschulen, die im Rahmen des Informatikunterrichts stattfinden, geben Schülerinnen und Schülern einen Einblick, welche konkreten Berufe in der IT-Branche gewählt werden können. Die Schülerinnen und Schüler lernen in der Kursstufe mit der objektorientierten Modellierung und Programmierung sowie Projektmanagement wichtige Fachkonzepte der Softwareentwicklung kennen. Sie erhalten dadurch die Möglichkeit, die verschiedenen Rollen und beruflichen Fachrichtungen in diesem elementaren Teilbereich der IT-Branche zu erleben und eine berufliche Zielrichtung zu entwickeln. Auch der Informatikunterricht der Kursstufe kann einen geschlechtsneutralen Zugang zur Arbeitswelt aufzeigen und stereotype Sichtweisen von informatiknahen Berufsbildern ausräumen.

- **Medienbildung (MB)**

Der reflektierte Umgang mit Medien verlangt ein grundlegendes Wissen über informatische Konzepte. Die Fachkompetenzen, die Schülerinnen und Schüler im Informatikunterricht erwerben, ermöglichen ihnen, eine Bewertung von Medien vorzunehmen und sie dadurch sicher und zielgerichtet einzusetzen. Im neuen Kompetenzbereich „Projektmanagement“ arbeiten die Schülerinnen und Schüler der Kursstufe intensiv mit digitalen Werkzeugen auf Softwarebasis, zum Beispiel Versionsverwaltungsprogrammen. Im Alltag sind Medien zum steten Begleiter geworden. Die im Informatikunterricht erlernten Strategien zum verantwortungsvollen Umgang mit Medien helfen den Schülerinnen und Schülern dabei, dass sie die Medien und nicht die Medien sie kontrollieren.

• **Verbraucherbildung (VB)**

In der Kursstufe lernen die Schülerinnen und Schüler mit Datenbanksystemen und Rechnernetzen spezielle Informatiksysteme kennen, die Grundlage für den Kauf und Verkauf von Waren und Dienstleistungen, für das Schalten von personalisierter Werbung und für die Unterbreitung persönlicher Angebote sind. Mit diesem Fachwissen werden die Schülerinnen und Schüler zu mündigen Bürgerinnen und Bürgern, die reflektiert Konsumententscheidungen treffen. Die Kenntnis darüber, wie personenbezogene Daten zur Ableitung weiterer Informationen über Personen genutzt werden, versetzt die Schülerinnen und Schüler in die Lage, Angebote und Dienstleistungen von Unternehmen kritisch zu beurteilen. Ausgehend von dieser Reflexionskompetenz entwickeln sie die Handlungskompetenz, konkrete Maßnahmen zum Schutz ihrer Daten zu ergreifen.

1.2 Kompetenzen

Die Auswirkung der Digitalisierung auf gesellschaftliche Entwicklungen hat in den vergangenen Jahren stetig zugenommen. Daher ist die Befähigung der Schülerinnen und Schüler, ihr Leben in einer Informationsgesellschaft selbstbestimmt führen und gestalten zu können und auch auf zukünftige Entwicklungen und die damit verbundenen Fragestellungen vorbereitet zu sein, nur durch den Erwerb entsprechender Kompetenzen erreichbar.

Zentrale Konzepte der Informatik

Grundlage für die Ausweisung von Kompetenzen sind zentrale Konzepte der Informatik. Dabei nehmen Konzepte des Informatischen Denkens (Computational Thinking) einen großen Anteil ein. Diese beschreiben den Prozess, ein Problem und die zur Verfügung stehenden Daten zu untersuchen, spezifische Muster zu erkennen, Wesentliches von Unwesentlichem zu unterscheiden und damit eine Lösung zu entwickeln, die so präzise beschrieben wird, dass sie leicht immer wieder ausgeführt werden kann. Wichtige Lösungsstrategien sind „Zerlegung in Teilprobleme“, „Abstrahieren“, „Mustererkennung“ und „Algorithmisierung“. An den Prozess der Problemlösung schließen sich Reflexion und Bewertung der Ergebnisse an. Diese Vorgehensweisen sind typisch für die Informatik, können aber auch in anderen Disziplinen angewendet werden. Die im Bildungsplan formulierten Kompetenzen stellen die Umsetzung dieser Konzepte im Informatikunterricht dar. Diese sind in zwei Bereiche unterteilt:

- Prozessbezogene Kompetenzen
- Inhaltsbezogene Kompetenzen

Ein zeitgemäßer Informatikunterricht berücksichtigt dabei stets die Verknüpfung von inhaltsbezogenen und prozessbezogenen Kompetenzen.



Zentrale Konzepte der Informatik in den prozessbezogenen und inhaltsbezogenen Kompetenzen (Grafik: Alexander Mittag)

Prozessbezogene Kompetenzen

Prozessbezogene Kompetenzen beschreiben im Bildungsplan Kompetenzen, die über alle Schuljahre eines Faches in einem längeren Prozess erworben werden. Die seit dem Aufbaukurs Informatik in Klasse 7 angebahnten prozessbezogenen Kompetenzen werden in der Kursstufe nun abschließend erworben. Die prozessbezogenen Kompetenzen gliedern sich in vier Kompetenzbereiche:

1. Strukturieren und Vernetzen
2. Modellieren und Implementieren
3. Kommunizieren und Kooperieren
4. Analysieren und Bewerten

Strukturieren und Vernetzen

Die Informatik als Strukturwissenschaft beschäftigt sich mit der Strukturierung von Daten und Prozessen (Algorithmen). Große Datenmengen können nur dann automatisiert und effizient verarbeitet werden, wenn sie in einer geeigneten Struktur vorliegen. Auch Algorithmen sind letztendlich Strukturen aus elementaren Bausteinen. Komplexere Problemstellungen können in einzelne Teilprobleme aufgeteilt werden, die oft für sich einfacher lösbar sind sowie Übersichtlichkeit und Wiederverwendbarkeit erhöhen. Die einzelnen Handlungsschritte werden anschließend chronologisch geordnet und zu einer Gesamtlösung vereint.

Modellieren und Implementieren

Um reale oder konstruierte Probleme lösen zu können, müssen sie zunächst aufbereitet werden. So können Details weggelassen werden, wenn sie für die Lösung irrelevant sind, oder sie müssen weggelassen werden, um ein Problem überhaupt beherrschbar zu machen. In den zur Verfügung stehenden Informationen müssen Regelmäßigkeiten, Wiederholungen, Ähnlichkeiten oder Gesetzmäßigkeiten erkannt werden, um charakteristische und verallgemeinerbare Bestandteile zu abstrahieren. Danach werden Abläufe, Daten und Beziehungen in informatischen Modellen dargestellt. Die Schülerinnen und Schüler implementieren Algorithmen in einer geeigneten Programmierumgebung und testen ihre Programme auf Fehler und die Ergebnisse auf Realitätsrelevanz.

Kommunizieren und Kooperieren

Die Schülerinnen und Schüler dokumentieren ihre Arbeitsschritte und (Teil-)Ergebnisse und bedienen sich dabei fachlicher Terminologie und geeigneter Visualisierungen. Nur dann können Lösungen beziehungsweise Lösungswege von anderen nachvollzogen werden. Sie bearbeiten geeignete Problemstellungen arbeitsteilig und verwenden dabei vorhandene Infrastruktur zur Kommunikation und Zusammenarbeit.

Analysieren und Bewerten

Die Analyse von Sachverhalten findet im Informatikunterricht auf unterschiedlichen Ebenen statt. So analysieren die Schülerinnen und Schüler Aufgabenstellungen, vorliegenden Programmcode, das Verhalten von Systemen mit unbekanntem inneren Aufbau (black box) sowie die gesellschaftlichen Auswirkungen von informatischen Systemen. Mit der Lösung eines Problems ist der Arbeitsprozess in der Informatik in der Regel nicht abgeschlossen. Beim Reflexionsprozess werden Lösungen mit der Ausgangssituation verglichen und gegebenenfalls Überlegungen zur Verbesserung angestellt. Dies führt zur Bewertung und Überarbeitung der Lösungen. In der Regel gibt es nicht nur eine richtige Lösung, sondern eine Vielzahl möglicher Umsetzungen. Darüber hinaus bewerten die Schülerinnen und Schüler die Auswirkung informatischer Anwendungen, Strukturen und Denkweisen auf die Gesellschaft sowie deren Sinnhaftigkeit.

Inhaltsbezogene Kompetenzen

Informatik beschäftigt sich mit der Darstellung, der automatischen Verarbeitung, Speicherung und Übertragung von Informationen. Dabei ist die Repräsentation der Information in Form von digitalen Daten Voraussetzung für deren weitere automatisierte Verarbeitung. Diese Prinzipien sind die Grundlage für die Gliederung der inhaltsbezogenen Kompetenzen.

Daten und Codierung

Codierungsvorschriften beschreiben, wie Informationen in ein standardisiertes Format gebracht werden können. Unterschiedliche Anwendungen stellen jeweils andere Anforderungen an die Codierungen. Die Schülerinnen und Schüler vervollständigen Ihre Kenntnisse über die Codierung von Zahlen und Texten und lernen weitere verlustfreie Verfahren zur Reduktion des Speicherbedarfs kennen. Ausgehend von Beispielen aus dem Alltag lernen die Schülerinnen und Schüler unterschiedliche Ansätze kennen, Daten mithilfe von Arrays, Listen, Bäumen und Graphen systematisch zu strukturieren. Das Konzept, interne Datenstrukturen zu kapseln, führt auf die abstrakten Datentypen Stack und Queue.

Relationale Datenbanksysteme ermöglichen die Organisation und Speicherung großer Datenmengen auf Basis des relationalen Datenmodells, bei dem Entitäten und ihre Beziehungen in Tabellen abgebildet werden. Mithilfe der Datenbanksprache SQL formulieren die Schülerinnen und Schüler Anfragen, um Informationen zu ermitteln.

Algorithmen

Die Schülerinnen und Schüler erweitern ihre vorhandenen Programmierkenntnisse. Sie lernen Standardalgorithmen aus verschiedenen Bereichen kennen und entwickeln und implementieren eigene Algorithmen. Mit der Objektorientierten Programmierung lernen die Schülerinnen und Schüler ein neues Programmierparadigma kennen. Sie implementieren eigene Klassen und nutzen das Geheimnisprinzip, um den sicheren Zugriff auf Attribute und Methoden zu regeln. Sie nutzen Assoziation und Vererbung zur Modellierung und Implementierung von Klassenbeziehungen. Algorithmen zur Verarbeitung größerer Datenmengen setzen voraus, dass die Daten in einer geeigneten Struktur vorliegen. Für viele Algorithmen spielen dabei die Grundprobleme Suchen, Traversieren und Sortieren eine Rolle. Für ein und dasselbe Problem gibt es oftmals verschiedene Lösungsverfahren, die sich in Speicher- und Zeitkomplexität sowie – bei Näherungsverfahren – in der Genauigkeit der Lösung unterscheiden. Die Schülerinnen und Schüler ergänzen ihre Problemlösestrategien um rekursive Algorithmen und erkennen deren Relevanz, indem sie konkrete Problemstellungen lösen.

Rechner und Netze

Anhand des Von-Neumann-Rechners lernen die Schülerinnen und Schüler eine konkret realisierbare Maschine und ihren inneren Aufbau kennen. Er stellt andererseits eine von der konkreten Realisierung unabhängige Modellmaschine dar, bei der Daten und Programm im selben Speicher liegen und deren Funktionalität auf einem ständigen Befehlszyklus basiert.

Die Vernetzung von Rechnern bildet die Grundlage verteilten Arbeitens und moderner Kommunikation. Dabei sind sowohl Konzepte der Datenübertragung als auch das Zusammenspiel der einzelnen Komponenten entscheidend.

Informationsgesellschaft und Datensicherheit

In der Informationsgesellschaft muss jeder Einzelne die Verantwortung für seine Daten im Hinblick auf Verfügbarkeit, Vertraulichkeit und Integrität übernehmen. Schülerinnen und Schüler nutzen im Alltag Technologien, denen kryptologische Systeme zugrunde liegen. Sie lernen den prinzipiellen Unterschied zwischen symmetrischen und asymmetrischen Verfahren kennen und erläutern, wie Infrastrukturen zur verschlüsselten Kommunikation genutzt werden. Sie wenden geeignete Maßnahmen zur Gewährleistung von Datensicherheit und Datenschutz an.

Automaten und formale Sprachen

Im Alltag begegnen den Schülerinnen und Schülern Phänomene, die sich mithilfe formaler Sprachen und Automatenmodellen beschreiben lassen. Ausgehend von diesen Phänomenen entwickeln die Schülerinnen und Schüler im Leistungsfach eine Vorstellung von formalen Sprachen, erwerben Kompetenzen im Umgang mit diesen und nutzen dabei unterschiedliche Darstellungsformen (zum Beispiel Automaten, Syntaxdiagramme, Produktionsregeln). In Basisfach und Leistungsfach nutzen die Schülerinnen und Schüler Automatenmodelle sowohl zur Modellierung realer Automaten (zum Beispiel Leergutautomaten, Snackautomaten und Ticketautomaten) als auch zur Modellierung zustandsbasierter Softwaresysteme.

Projektmanagement

Komplexere Projekte lassen sich heute nur noch in Teams planen und realisieren. Zur Planung und Durchführung haben sich verschiedene Methoden etabliert. Die Schülerinnen und Schüler lernen Vorgehensmodelle und gängige Begriffe des Projektmanagements kennen. Bei der Realisierung eigener Projekte mit dem Ziel, funktionsfähige Prototypen zu erstellen, wenden sie Methoden des Projektmanagements von der Planung über die Durchführung bis zur Dokumentation an.

Nummerierung von Teilkompetenzen in Basis- und Leistungsfach

Zu besserer Übersichtlichkeit sind die Teilkompetenzen in Basisfach und Leistungsfach gleich durchnummeriert. Teilkompetenzen, die nur im Leistungsfach auftauchen, sind im Basisfach mit „[nur LF]“ gekennzeichnet.

1.3 Didaktische Hinweise

Vorkenntnisse

Aufbauend auf den Grundlagen des Informatikunterrichts aus Klasse 7 können in der Mittelstufe informatische Kompetenzen auf unterschiedlichen Wegen erworben werden. Dabei werden die Kompetenzen der prozessbezogenen und der inhaltsbezogenen Kompetenzbereiche in spiralcurricularer Weise aufgebaut. Der Zugang zu Basisfach und Leistungsfach Informatik ist damit für Schülerinnen und Schüler aus unterschiedlichen Richtungen geöffnet.

Differenzierung

Grundlegende Konzepte der Informatik stehen im Mittelpunkt des Unterrichts. Während in Klasse 7 viele Begriffe und Zusammenhänge zunächst propädeutisch behandelt werden und sich die Ausformulierungen der Niveaustufen einiger Teilkompetenzen nur geringfügig unterscheiden, findet in der Mittelstufe zunehmend eine Systematisierung und Vernetzung der Konzepte statt. Mit ansteigender Komplexität der Inhalte steigt auch die Notwendigkeit der Differenzierung im Unterricht. Dies kann beispielsweise über unterschiedlich komplexe Beispiele, unterschiedliche Anforderungsniveaus, Aufgabenvariationen oder gestufte Hilfen erfolgen.

Variation von Problemstellungen

In besonderer Weise bietet der informatische Zugang zur Problemlösung auch die Möglichkeit, über die Grenzen der ursprünglichen Aufgabenstellung hinaus zu denken. Durch veränderte Anforderungen, Rahmenbedingungen oder Variation der Problemgröße werden informatische Lösungskonzepte verdeutlicht, nachvollziehbar gemacht oder hinterfragt. Fragestellungen der Art „Wie hätte man es anders machen können?“, „Ist die gefundene Lösung optimal?“ oder „Wo enden die Einsatzbereiche der gefundenen Lösung?“ gehören zum Standardrepertoire des Informatikunterrichts. So entwickelt sich der Unterrichtsgegenstand im Informatikunterricht von anfänglich konkreten Problemstellungen hin zu ganzen Problemklassen.

Geschlechtsneutraler Informatikunterricht

Schülerinnen und Schüler besitzen oftmals diffuse Vorstellungen von Informatik und deren Aufgabenbereichen und bringen Rollenklischees mit, die sich dann im Unterricht widerspiegeln. Diese Problematik wird im Informatikunterricht aufgegriffen: Programmierbeispiele und Übungen werden so gewählt, dass sie in gleichem Maße Jungen und Mädchen ansprechen. Bei der Problemlösung werden schülerspezifische Herangehensweisen berücksichtigt, planvolle und probierende Vorgehensweisen ergänzen sich sinnvoll.

Aktuelle Bezüge

Schülerinnen und Schüler werden in alltäglichen Situationen oder in den Nachrichten mit informatischen Themen konfrontiert. Dies reicht von zunächst unerklärbarem Verhalten (zum Beispiel Fehlermeldungen) beim Bedienen von Endgeräten bis zu Nachrichten über technische Entwicklungen oder Zwischenfälle im Zusammenhang mit Informatiksystemen. Viele dieser aktuellen Ereignisse bieten geeignete Anknüpfungsmöglichkeiten, um sowohl die informatisch-technologischen Aspekte als auch die Auswirkungen auf Individuum und Gesellschaft zu beleuchten.

Informatik in Industrie und Berufswelt

Fast alle Elemente entlang einer Wertschöpfungskette sind zunehmend miteinander vernetzt. Daher weisen immer mehr Berufsfelder informatische Bezüge auf und für viele Berufe ist inzwischen ein grundlegendes Verständnis informatischer Konzepte notwendig. An geeigneten Beispielen von Industriezweigen oder Berufsbildern können Schülerinnen und Schüler dies kennenlernen.

Programmieren

Programmieren als Realisierung von Ideen in Software als schöpferischer und produktiver Prozess ist ein wesentlicher Bestandteil des Informatikunterrichts. Die Schülerinnen und Schüler entwerfen Problemlösungen, die auf grundlegenden Programmierbausteinen basieren und erfahren so, dass die Lösung nicht in den Bausteinen selbst, sondern hauptsächlich in der Art und Weise ihrer Anordnung liegt. Grundsätzlich bieten Programmieraufgaben die Chance, dass Schülerinnen und Schüler die Arbeitsergebnisse anhand des Programmablaufs beziehungsweise -ergebnisses selbstständig und unabhängig von der Lehrkraft überprüfen können. Hier ist eine behutsame Heranführung durch die Lehrkraft erforderlich, damit die Schülerinnen und Schüler lernen, diese objektive Rückmeldung zur Weiterentwicklung ihrer Lösung zu nutzen.

Programmierungsumgebung

Die Entscheidung für eine geeignete Programmiersprache beziehungsweise Programmierungsumgebung sollte stets nach Gesichtspunkten der altersangemessenen Vermittlung informatischer Konzepte erfolgen.

Zeitgemäße Unterstützungsfunktionen (wie zum Beispiel Codevervollständigung, Refactoring, Syntaxhervorhebung, Statische Codeanalyse, UML, Debugging, Einbinden von Bibliotheken und Buildwerkzeugen, Testen, Kollaboration, Versionierung) werden eher durch professionelle Entwicklungsumgebungen unterstützt. Aufgrund der Funktionsvielfalt erschweren diese jedoch häufig den Blick für die wesentlichen im Informatikunterricht zu vermittelnden Konzepte.

Dem gegenüber stehen Programmierungsumgebungen, die durch spezielle didaktische Funktionen (zum Beispiel Objektinteraktion, Visualisierungen) und reduzierten Funktionsumfang Programmieranfängern den Einstieg erleichtern und das Verständnis für die zugrundeliegenden Konzepte schärfen, jedoch möglicherweise bei wachsenden Anforderungen an ihre Grenzen stoßen.

Im Bewusstsein des Spannungsfeldes zwischen professionellen und didaktischen Entwicklungsumgebungen sollte in unterschiedlichen Kontexten die jeweils dafür geeignete Entwicklungsumgebung ausgewählt werden.

Grafische Benutzerschnittstelle

Die Möglichkeit einer ansprechenden grafischen Benutzerschnittstelle (GUI) hat für Schülerinnen und Schüler einen hohen Aufforderungscharakter und motiviert dazu, eigene Ideen umzusetzen. Dennoch muss der Lehrkraft bewusst sein, dass eine umfassende Beschäftigung mit der Thematik „grafische Benutzerschnittstelle“ aufgrund der damit zwangsweise verbundenen Unterthemen (zum Beispiel Eventhandling, Skalierung, User Experience, Design, Interaktivität, MVC etc.) zeitlich nicht leistbar ist.

Informatik mit und ohne Rechnereinsatz

Die Beschäftigung mit informatischen Inhalten geschieht nicht ausschließlich mit digitalen Endgeräten. Die Vermittlung mithilfe von „unplugged“-Elementen (zum Beispiel Nachspielen mit geeigneten Gegenständen, in Papierform, durch Rollenspiele) kann dazu beitragen, die Kernidee informatischer Konzepte in den Vordergrund zu rücken und Ablenkungen, die im Umgang mit Werkzeugen oft unvermeidlich sind, zu reduzieren. Eine Reihe von didaktischen Werkzeugen ermöglicht es jedoch auch, am Computer informatische Konzepte erlebbar zu machen, zum Beispiel bei der Modellierung, Simulation und Visualisierung von Netzwerken, logischen Schaltungen, Rechnermodellen, Datenbanken, Automaten, Objekten und ihren Beziehungen untereinander.

Projektmanagement

Der Informatikunterricht bietet den Schülerinnen und Schülern handlungs- und problemorientierte Unterrichtssituationen, in denen sie erworbene Kompetenzen und erlerntes Wissen anwenden können. In der Mittelstufe haben die Schülerinnen und Schüler schon erste Erfahrungen mit kleineren Programmierprojekten gesammelt. In der Kursstufe wird der Projektbegriff nun fachlich untermauert, indem die Schülerinnen und Schüler verschiedene Aspekte und Methoden des Projektmanagements kennenlernen und diese bei der Umsetzung eigener Softwareprojekte anwenden. Dabei erwerben sie weitere wichtige Kompetenzen wie das Erstellen von Dokumentationen sowie das Auffinden semantischer Fehler und Behandeln von Laufzeitfehlern.

Sächliche Ausstattung und Infrastruktur

Bei der Konzeption des Unterrichts sind stets die örtlichen Gegebenheiten (unter anderem räumliche Lösung, Endgeräte, Infrastruktur) zu berücksichtigen. Grundsätzlich ist der Einsatz digitaler Endgeräte selbstverständlich. Der Einsatz von Software, Lernplattformen etc. muss dabei individuell auf vorhandene Strukturen und Ausstattung abgestimmt werden. Dem Umgang mit der Infrastruktur kommt dabei die Rolle eines Werkzeuges zu. Dieses soll gegenüber der Vermittlung informatischer Inhalte in den Hintergrund treten. Wie Werkzeuge in der einzelnen Unterrichtsstunde eingesetzt werden, ist in der jeweiligen didaktischen Konzeption zu überprüfen.

Verschränkung der Bereiche

Die anfänglich noch eher singulär betrachteten inhaltsbezogenen Kompetenzbereiche werden zunehmend miteinander vernetzt. Komplexere reale Informatiksysteme basieren fast immer auf Inhalten aus allen inhaltsbezogenen Kompetenzbereichen. Dem soll im Informatikunterricht durch geeignete Anwendungsbeispiele Rechnung getragen werden, bei denen diese Verschränkung deutlich wird. Dies könnte aufgezeigt werden, indem für eine entwickelte Anwendung beispielsweise noch eine Datenbankschnittstelle implementiert wird, das Modell eines endlichen Automaten zur Realisierung einer interaktiven GUI genutzt wird, Datenstrukturen und die darauf operierenden Algorithmen rekursiv betrachtet werden oder Dateiformate als formale Sprache betrachtet werden.

1.4 Wege in die Kursstufe über Profulfach IMP oder Brückenkurs

Während Informatik in Klasse 7 in allen Schularten als Pflichtfach unterrichtet wird, gibt es am Gymnasium zwei Wege, die informatischen Kompetenzen zu erwerben, die zum Besuch von Informatik als Basisfach oder Leistungsfach in der gymnasialen Kursstufe berechtigen.

Im Profulfach IMP werden informatische Kompetenzen mit einem Anteil von insgesamt 4 Jahreswochenstunden über drei Schuljahre hinweg erworben. Besondere Berücksichtigung finden hier unter anderem projektartiges Arbeiten und die Verknüpfung mit Kompetenzen aus den Bereichen Mathematik und Physik. Der Brückenkurs umfasst hingegen nur zwei Jahreswochenstunden (einstündig in den Klassenstufen 9 und 10 oder zweistündig in Klasse 10). Ein Fokus des Brückenkurses liegt auf der Vorbereitung für die gymnasiale Kursstufe, weshalb im Vergleich zu IMP an vielen Stellen stringenter vorgegangen wird und kleinere Projekte oder Querverknüpfungen zu anderen Bereichen im Vergleich zum Profulfach IMP deutlich kürzer ausfallen oder – sofern die Kompetenzen für den Besuch der Kursstufe nicht zwingend notwendig sind – nicht vorhanden sind.

In Klasse 11 der Kursstufe werden somit möglicherweise Schülerinnen und Schüler mit unterschiedlichen Bildungswegen (Profulfach IMP und Brückenkurs) in einem gemeinsamen Kurs sitzen, wie dies beispielsweise auch bei den modernen Fremdsprachen der Fall ist. Diesem Umstand muss insbesondere beim Einstieg in neue Themen oder Unterrichtseinheiten Rechnung getragen werden.

2. Prozessbezogene Kompetenzen

2.1 Strukturieren und Vernetzen

Die Schülerinnen und Schüler ordnen Objekte auf verschiedene Art und Weise an. Sie lernen verschiedene Strukturen zur Vernetzung von Daten (Liste, Baum, Graph) und deren Einsatzmöglichkeiten kennen – sowohl zur Problemlösung als auch im Arbeitsalltag. Sie erfahren, dass sinnvoll strukturierte Daten zum (schnellen) Wiederauffinden unerlässlich sind und erst eine effiziente automatische Verarbeitung ermöglichen.

Sie verwenden textuelle und grafische Modellierungssprachen, unter anderem zum Beschreiben von Objekten, Beziehungen, Abhängigkeiten, Grammatiken, Protokollen, Prozessen oder Strukturen.

Die Strukturierung ihrer eigenen Arbeitsprozesse innerhalb von Projekten ermöglicht zielgerichtete und fehlertolerante Zusammenarbeit.

Die Schülerinnen und Schüler können

Daten strukturieren und vernetzen

1. mit dem Schulnetz (zum Beispiel Homeverzeichnis, Tauschverzeichnis, mobile Datenträger, Netzwerkdrucker) zielorientiert arbeiten
2. Dateien und Bezeichner (zum Beispiel für Variablen, Unterprogramme) aussagekräftig benennen
3. Beziehungen zwischen Daten/Objekten (zum Beispiel Hierarchien in Verzeichnisbäumen oder Stammbäumen, die Struktur des Internets, Verkehrsnetz als Graph) erkennen und erläutern
4. gleichartige Daten in geeigneten Datenstrukturen zusammenfassen (zum Beispiel Namensliste einer Klasse, Pixel einer Rastergrafik etc.)

Prozesse strukturieren und vernetzen

5. Handlungsschritte chronologisch ordnen (auch aufgrund von kausalen Zusammenhängen)
6. Teillösungen zur Lösung des Gesamtproblems nutzen
7. Schnittstellen für Teilbereiche definieren, die unabhängig voneinander bearbeitet werden (zum Beispiel Gruppenarbeit, Protokolle bei Client-Server, Parameter und Rückgabewerte bei Unterprogrammen)
8. textuelle und grafische Modellierungssprachen zur Beschreibung und Entwicklung von Strukturen und Prozessen verwenden

Eigene Arbeitsprozesse strukturieren

9. ihre eigenen Arbeitsprozesse in Projekten planen und strukturieren, auf Abweichungen reagieren und den Verlauf dokumentieren

2.2 Modellieren und Implementieren

Die Schülerinnen und Schüler können Problemstellungen sowohl der realen Welt als auch aus konstruierten Problemstellungen aufbereiten und daraus informatische Modelle erstellen, diese in einer geeigneten Umgebung implementieren, ihre korrekte Funktionsfähigkeit testen und so funktionsfähige informatische Systeme kreieren.

Sie entwickeln Programme zur Problemlösung. Ausgehend von spielerisch-probierenden Ansätzen gehen sie dabei zunehmend planvoll und strukturiert vor. Sie können Strategien zum Problemlösen auswählen, ihre Auswahl begründen und daraus unter Verwendung von geeigneten Zwischenschritten und/oder Ideenskizzen einen Plan zur Lösung entwickeln. Systematisches Testen, Fehlersuche und Verifizieren eines Ergebnisses sind dabei zunehmend feste Bestandteile des Implementierungsprozesses. Sie untersuchen, inwieweit die Umsetzung den Erfordernissen der Aufgabenstellung/Realsituation entspricht.

Die Schülerinnen und Schüler lernen, Problemstellungen zunehmend in verschiedenen Abstraktionsschichten zu betrachten.

Die Schülerinnen und Schüler können

Problemstellungen analysieren und aufbereiten

1. die für die Problemstellung relevanten Informationen herausarbeiten und fehlende beziehungsweise ergänzende Informationen beschaffen
2. für (Teil-)Abläufe notwendige Eingabedaten und Ergebnisse beschreiben und in Form von Testfällen formalisieren
3. vorliegende Informationen für die Lösung geeignet aufbereiten (zum Beispiel durch Filtern, Reduktion, Kategorisieren)
4. charakteristische und verallgemeinerbare Bestandteile herausarbeiten (Abstraktion)

Konzipieren und Lösungen entwickeln

5. relevante Abläufe, Daten, Komponenten und ihre Beziehungen in informatischen Modellen darstellen
6. passende Komponenten, Strukturen und Lösungsstrategien für gegebene Problemstellungen auswählen und adaptieren
7. geeignete Programme und Hilfsmittel zur grafisch gestützten Modellierung einsetzen
8. unterschiedliche Perspektiven in die Entwicklung einer Lösung miteinbeziehen

Implementieren

9. Strukturen und Abläufe in einer Programmiersprache implementieren
10. geeignete Codebausteine aus verschiedenen Quellen auswählen, gegebenenfalls adaptieren und in eigene Programme einbauen und dabei rechtliche Rahmenbedingungen berücksichtigen

Testen und reflektieren

11. Programme gezielt gegen vorab formulierte Testfälle testen
12. Fehler in der Implementierung systematisch aufspüren und beheben
13. die Angemessenheit von Lösungen und die erreichten Resultate bewerten

2.3 Kommunizieren und Kooperieren

Die Schülerinnen und Schüler erwerben die Fähigkeiten, um informatische Sachverhalte zunehmend unter Verwendung von Fachsprache zu diskutieren. Sie dokumentieren ihre Ideen, Beobachtungen, Lösungswege und (Teil-)Ergebnisse und verwenden geeignete Medien und (fachspezifische) Notationsweisen zur Visualisierung.

Die Schülerinnen und Schüler nutzen vorhandene Medien und Infrastruktur zur Kommunikation und Kooperation. Sie präsentieren technische Sachverhalte, Arbeitsprozesse und Ergebnisse in geeigneter Form und verwenden dabei eine wertschätzende und geschlechtersensible Sprache.

Sie setzen sich kritisch mit Fragen zum Spannungsfeld zwischen Informatik und Gesellschaft auseinander und beachten in ihrer Arbeitsweise erste rechtliche Aspekte. Dabei zeigen sie einen respektvollen Umgang und Offenheit gegenüber anderen Lösungswegen, Meinungen und Ansichten und diskutieren Aspekte von Toleranz und Akzeptanz von Vielfalt im Kontext informatischer Fragestellungen.

Die Schülerinnen und Schüler können

Überlegungen, Lösungswege und Ergebnisse darstellen

1. fachspezifische Schreib- und Notationsweisen verwenden
2. Sachverhalte, eigene Ideen, Lösungswege und Ergebnisse zielgruppenorientiert und unter Beachtung der informatischen Terminologie erläutern und strukturiert darstellen

Dokumentieren und kommentieren

3. eigenen und fremden Programmcode in geeigneter Weise kommentieren und dokumentieren
4. vorhandene Dokumentationen und kommentierten Programmcode lesen und verstehen

Kooperativ arbeiten

5. arbeitsteilig als Team ihre Aufgaben planen, strukturieren, ausführen, reflektieren und präsentieren
6. zielorientiert auf einer vorhandenen Infrastruktur kommunizieren und geeignete digitale Werkzeuge zum kollaborativen Arbeiten verwenden

Kommunizieren in der Gesellschaft

7. in Erarbeitung, Kooperation und Darstellung alltagsrelevante rechtliche Regelungen befolgen und verantwortungsvoll mit eigenen und fremden personenbezogenen Daten umgehen
8. charakteristische Merkmale verschiedener Kommunikationsformen (Mensch-Mensch, Mensch-Maschine, Maschine-Maschine) auf Gemeinsamkeiten und Unterschiede analysieren und deren gesellschaftliche Auswirkungen bewerten
9. Sicherheitsaspekte bei ihrem Kommunikationsverhalten berücksichtigen und die gesellschaftliche Relevanz von verschlüsselter Kommunikation reflektieren
10. Aspekte von Toleranz und Akzeptanz von Vielfalt im Kontext informatischer Fragestellungen diskutieren

2.4 Analysieren und Bewerten

Die Schülerinnen und Schüler untersuchen eigene und gegebene Programme und informatische Systeme. Die Analyse von Code führt dabei, ausgehend von der Identifikation der verwendeten Kontrollstrukturen, über ein schrittweises Nachvollziehen des Programmablaufs zum Begreifen der Funktionalität des Programms. Ihr Wissen über die innere Struktur von Informatiksystemen befähigt sie, Risiken und Chancen einzuschätzen und gegebenenfalls geeignete Sicherheitsmaßnahmen zu ergreifen. Dabei berücksichtigen sie sowohl technische und sicherheitsrelevante als auch gesellschaftliche und ethische Aspekte.

Die Schülerinnen und Schüler können

Informatische Aspekte

1. durch Analyse (zum Beispiel „gezieltes Anwenden“/Blackbox oder auch Codebetrachtung/Whitebox) Erkenntnisse über das Verhalten von informatischen Systemen gewinnen
2. informatische Modelle mit der jeweiligen Realsituation vergleichen
3. unterschiedliche Lösungsansätze und Vorgehensweisen miteinander vergleichen und bewerten
4. Optimierungsbedarf ermitteln und gegebenenfalls Lösungswege optimieren
5. Kenntnisse über den inneren Ablauf informatischer Systeme im Alltag nutzen
6. Einsatzbereiche und Grenzen von Modellen erkennen
7. Entscheidungen auf der Grundlage informatischen Sachverstands treffen und diese sachgerecht begründen

Gesellschaftliche Aspekte

8. Auswirkungen von Computersystemen auf Gesellschaft, Berufswelt und persönliches Lebensumfeld aus verschiedenen Perspektiven bewerten
9. im Zusammenhang einer digitalisierten Gesellschaft einen eigenen Standpunkt zu ethischen Fragen in der Informatik einnehmen und ihn argumentativ vertreten

3. Standards für inhaltsbezogene Kompetenzen

3.1 Klassen 9/10 (Brückenkurs)

3.1.1 Daten und Codierung

Die Schülerinnen und Schüler lernen mit dem Hexadezimalsystem ein weiteres System zur Kodierung von Zahlen kennen.

Zur effizienten Speicherung großer Datenmengen lernen die Schülerinnen und Schüler Verfahren zur verlustfreien und verlustbehafteten Kompression kennen. Anhand der Lauflängencodierung können die Schülerinnen und Schüler ein konkretes Verfahren explizit nachvollziehen und anwenden.

Die Schülerinnen und Schüler können

Klasse 9*	
(1)	die Überführung von Zahlen zwischen ihrer Darstellung im <i>Hexadezimalzahlssystem</i> , <i>Binärsystem</i> und <i>Dezimalsystem</i> durchführen
F	IMP 3.1.2.1 Mathematische Grundlagen der Kryptologie (2), (3)
L	MB Informationstechnische Grundlagen
(2)	Verfahren zur <i>Datenreduktion</i> beziehungsweise <i>verlustbehafteten Datenkompression</i> (zum Beispiel Verringerung von Farbtiefe, Auflösung, Samplingtiefe, Samplingrate) beschreiben
F	IMP 3.2.1.1 Daten und Codierung (2)
(3)	die <i>Lauflängencodierung</i> als Beispiel für ein <i>verlustfreies Datenkompressionsverfahren</i> erläutern und an einem Beispiel händisch durchführen (zum Beispiel Kompression von s/w-Bildern)
F	IMP 3.2.1.1 Daten und Codierung (3)
(4)	Einsatzbereiche und Beispiele für <i>verlustbehaftete</i> und <i>verlustfreie Datenkompressionsverfahren</i> nennen (zum Beispiel Kompression von Videos, Grafiken, Musik, Programmcode oder Textdokumenten)
F	IMP 3.2.1.1 Daten und Codierung (4)
L	MB Information und Wissen; Informationstechnische Grundlagen

* Falls das Fach in den Klassen 9 und 10 einstündig unterrichtet wird

3.1.2 Algorithmen

Die Schülerinnen und Schüler lernen die Syntax der schon bekannten algorithmischen Grundbausteine in einer textuellen Programmiersprache kennen.

Sie nutzen Variablen mit unterschiedlichen Datentypen und auch Arrays zur Speicherung und Verarbeitung gleichartiger Daten. Sie nutzen Unterprogramme mit Parametern und Rückgabewerten, beschreiben Anforderungen an Unterprogramme und entwerfen geeignete Testroutinen, um ihre Implementierungen dagegen zu testen. Sie unterscheiden zwischen syntaktischen und semantischen Fehlern. Sie entwickeln Strategien, um fehlerfreien Code zu schreiben und nutzen auch die Möglichkeiten der Entwicklungsumgebung.

Der Fokus liegt auf den grundlegenden Sprachelementen einer textuellen Programmiersprache. Für spezielle Aufgabenbereiche (zum Beispiel Ein- und Ausgabe von Text, Grafik, Sensorabfragen) können geeignete Programmbibliotheken zum Einsatz kommen.

Sie implementieren interaktive Programme mit einer einfachen grafischen Benutzerschnittstelle.

Die Schülerinnen und Schüler können

Klasse 9*	
(1)	<i>Algorithmen</i> mit den Grundbausteinen <i>Anweisung</i> , <i>Bedingung</i> , <i>Schleife</i> und <i>Verzweigung</i> sowie unter Verwendung von <i>Variablen</i> in einer geeigneten textuellen Programmiersprache implementieren
	<p>P 2.2 Modellieren und Implementieren 9</p> <p>F IMP 3.2.1.2 Algorithmen (1)</p>
(2)	die <i>Datentypen</i> für Ganzzahl, Gleitkommazahl, Wahrheitswert und Zeichenkette beschreiben und anwenden
	F IMP 3.2.1.2 Algorithmen (2)
(3)	logische Verknüpfungen (<i>UND</i> , <i>ODER</i> , <i>NICHT</i>) in Bedingungen von <i>Schleifen</i> und <i>Verzweigungen</i> verwenden und deren Wahrheitstafeln angeben
	F IMP 3.1.1.2 Algorithmen (1)
(4)	Zufallszahlen in eigenen Programmen verwenden
	<p>F IMP 3.2.1.2 Algorithmen (3)</p> <p>F M 3.3.5 Leitidee Daten und Zufall</p>
(5)	<i>Unterprogramme</i> – auch mit <i>Parametern</i> und <i>Rückgabewerten</i> – sinnvoll verwenden
	<p>P 2.1 Strukturieren und Vernetzen 2, 6, 7</p> <p>P 2.2 Modellieren und Implementieren 6</p> <p>F IMP 3.2.1.2 Algorithmen (6)</p> <p>L PG Selbstregulation und Lernen</p>
(6)	den Unterschied zwischen verschiedenen Fehlerarten (<i>Compilerfehler/Laufzeitfehler</i> und <i>syntaktisch/semantisch</i>) erläutern
	<p>F D 3.3.2.1 Struktur von Äußerungen</p> <p>F IMP 3.2.1.2 Algorithmen (9)</p>
(7)	vorgegebenen Code auf seine Funktionsweise hin analysieren und dessen Wirkung beschreiben
	<p>P 2.1 Strukturieren und Vernetzen 3</p> <p>P 2.2 Modellieren und Implementieren 4</p> <p>P 2.3 Kommunizieren und Kooperieren 4</p> <p>P 2.4 Analysieren und Bewerten 1</p> <p>F IMP 3.2.1.2 Algorithmen (11)</p> <p>L MB Medienanalyse</p>
(8)	Anpassungen zu vorgegebenem Code implementieren
	<p>P 2.1 Strukturieren und Vernetzen 3</p> <p>P 2.2 Modellieren und Implementieren 4, 6, 9, 10</p> <p>P 2.3 Kommunizieren und Kooperieren 4</p> <p>P 2.4 Analysieren und Bewerten 4</p> <p>F IMP 3.2.1.2 Algorithmen (12)</p>

Klasse 10*
(9) vorgegebene Programmbibliotheken für eigene Programme sinnvoll verwenden
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 4 F IMP 3.2.1.2 Algorithmen (15) L MB Kommunikation und Kooperation L PG Selbstregulation und Lernen
(10) <i>Algorithmen</i> entwerfen und implementieren, die Benutzereingaben anfordern und auswerten
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 5, 7 P 2.2 Modellieren und Implementieren 9 F IMP 3.2.1.2 Algorithmen (16)
(11) Programmcode implementieren, der Daten aus Dateien einliest, verarbeitet und in Dateien schreibt (mittels geeigneter Bibliotheken)
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 7 P 2.3 Kommunizieren und Kooperieren 4 F IMP 3.2.1.2 Algorithmen (17)
(12) <i>Arrays</i> zur Speicherung und Verarbeitung von Daten verwenden
<ul style="list-style-type: none"> F IMP 3.2.1.2 Algorithmen (4)
(13) grundlegende <i>Algorithmen</i> auf <i>Arrays</i> (zum Beispiel Füllen mit Werten, Maximumsuche, Summenbildung, Bubblesort) erläutern und implementieren
<ul style="list-style-type: none"> F IMP 3.2.1.2 Algorithmen (5)
(14) Strategien (zum Beispiel Debugger, schrittweise Ausführung, Logging) anwenden, um das Verhalten von Programmcode zur <i>Laufzeit</i> zu beobachten
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 12 P 2.4 Analysieren und Bewerten 1 F IMP 3.2.1.2 Algorithmen (14) L PG Selbstregulation und Lernen
(15) ein interaktives Programm (zum Beispiel App, Webanwendung, Desktopanwendung) mit einer einfachen <i>grafischen Benutzerschnittstelle</i> (zum Beispiel mit Buttons, Texteingabe und Ausgabe) implementieren, je nach Sprache unter Verwendung geeigneter (didaktischer) Toolkits und/oder GUI-Builder
<ul style="list-style-type: none"> F IMP 3.3.1.2 Algorithmen (3) L BO Einschätzung und Überprüfung eigener Fähigkeiten und Potenziale; Fachspezifische und handlungsorientierte Zugänge zur Arbeits- und Berufswelt L MB Produktion und Präsentation L PG Selbstregulation und Lernen L VB Qualität der Konsumgüter L MB Informationstechnische Grundlagen

* Falls das Fach in den Klassen 9 und 10 einstündig unterrichtet wird

3.1.3 Rechner und Netze

Die Schülerinnen und Schüler lernen den grundlegenden Aufbau eines lokalen Rechnernetzes sowie die Rolle der beteiligten Komponenten kennen. Bei der Modellierung mithilfe einer geeigneten Software werden sowohl Mechanismen wie Adressierung, Namensauflösung als auch das Prinzip des Domain Name Systems angesprochen. Die Schülerinnen und Schüler erstellen in einer Simulationsumgebung eigene Webseiten mit Links auch zu anderen Webservern.

Das Problem des Datentransports über mehrere Knoten (Routing) wird in einer geeigneten Simulationsumgebung nachgestellt und in verschiedenen Szenarien simuliert.

Die Schülerinnen und Schüler können

Klasse 10*	
(1)	den grundlegenden Aufbau eines <i>lokalen Rechnernetzes</i> und die Rolle seiner Komponenten (Endgerät, Verbindung, Verteiler) erklären
	F IMP 3.1.1.3 Rechner und Netze (1)
(2)	die Notwendigkeit einer eindeutigen <i>Adressierung</i> zur Kommunikation in Netzen erläutern und hierfür Beispiele nennen (<i>IP-Adresse</i> und zum Beispiel Handynummer, E-Mail-Adresse)
	F IMP 3.1.1.3 Rechner und Netze (2)
(3)	das Prinzip der <i>Namensauflösung</i> (<i>DNS</i> und zum Beispiel Kontaktliste, Telefonbuch) erläutern
	F IMP 3.1.1.3 Rechner und Netze (5)
(4)	ein <i>lokales Rechnernetz</i> mit <i>DNS</i> und <i>Webserver</i> in einer geeigneten Simulationsumgebung entwerfen und untersuchen
	P 2.2 Modellieren und Implementieren 5, 7 F IMP 3.1.1.3 Rechner und Netze (6)
(5)	Schemata beschreiben, mit denen eine Unterscheidung von <i>Adressen</i> in Netzwerken nach lokal/global möglich ist (zum Beispiel Subnetzmaske in IP-Netzen, Vorwahl im Telefonnetz, Länderkennung bei Postanschrift)
	F IMP 3.3.1.3 Rechner und Netze (7)
(6)	das Problem des <i>Routings</i> zwischen Netzen erläutern und in einer geeigneten Simulationsumgebung ein Routingszenario durchführen
	F IMP 3.3.1.3 Rechner und Netze (8)
	L MB Informationstechnische Grundlagen; Kommunikation und Kooperation

* Falls das Fach in den Klassen 9 und 10 einstündig unterrichtet wird

3.1.4 Informationsgesellschaft und Datensicherheit

Aus Klassenstufe 7 sind den Schülerinnen und Schülern die Cäsar-Verschlüsselung und ihre Schwachstellen bekannt. Anhand ihrer Weiterentwicklung, der Vigenère-Verschlüsselung, und deren Kryptoanalyse lernen die Schülerinnen und Schüler Sicherheitsaspekte von Verschlüsselungsverfahren kennen. Diese Aspekte führen sie auf das absolut sichere One-Time-Pad-Verfahren und auf das Konzept der asymmetrischen Verschlüsselung. In unserer heutigen Informationsgesellschaft sind personenbezogene Daten ein wertvolles Gut. Sie werden von verschiedenen Stellen automatisiert erhoben, zusammengeführt und ausgewertet. Das Verständnis der hierbei eingesetzten Technologien ist die Voraussetzung dafür, dass die Schülerinnen und Schüler geeignete Strategien entwickeln, um sensibel mit ihren persönlichen Daten umzugehen.

Die Schülerinnen und Schüler lernen, dass ein asymmetrisches Verfahren einen sicheren, unmittelbaren Schlüsselaustausch als Grundlage für eine sichere Kommunikation gewährleisten kann.

Die Schülerinnen und Schüler können

Klasse 10*	
(1)	das <i>Vigenère-Verfahren</i> erklären und durchführen
	F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (1)
(2)	<i>Transpositionsverfahren</i> (zum Beispiel Skytale), <i>monoalphabetische Substitution</i> und <i>polyalphabetische Substitution</i> vergleichen
	P 2.4 Analysieren und Bewerten 3 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (2)
(3)	eine grundlegende Angriffsstrategie auf das <i>Vigenère-Verfahren</i> erklären und an einfachen Beispielen durchführen
	F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (3)
(4)	das <i>One-Time-Pad-Verfahren</i> erklären und begründen, dass es sich um ein absolut sicheres Verschlüsselungsverfahren handelt
	P 2.4 Analysieren und Bewerten 7 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (4)
(5)	erläutern, dass die Sicherheit von Verschlüsselungsverfahren nicht von der Geheimhaltung des <i>Algorithmus</i> abhängen darf (<i>Kerckhoffs'sches Prinzip</i>)
	P 2.4 Analysieren und Bewerten 7 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (6)
(6)	Anwendungsbereiche beschreiben, in denen Verschlüsselung eingesetzt wird (zum Beispiel verschlüsselte Speicherung von Daten, Kommunikation über https oder Messenger)
	F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (7) L BO Fachspezifische und handlungsorientierte Zugänge zur Arbeits- und Berufswelt L BTV Minderheitenschutz; Personale und gesellschaftliche Vielfalt; Toleranz, Solidarität, Inklusion, Antidiskriminierung; Wertorientiertes Handeln L MB Jugendmedienschutz; Mediengesellschaft L PG Mobbing und Gewalt
(7)	eine aktuell eingesetzte Technologie erläutern, mit der personenbezogene Daten gesammelt werden (zum Beispiel Webtracking, Cookies, Geodaten)
	P 2.3 Kommunizieren und Kooperieren 2, 8, 9, 10 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (9) L BTV Minderheitenschutz; Wertorientiertes Handeln L VB Chancen und Risiken der Lebensführung
(8)	Möglichkeiten erläutern, um das Sammeln <i>personenbezogener Daten</i> einzuschränken (zum Beispiel anonymes Surfen, Rechteverwaltung von Apps, Standortfreigabe)
	P 2.3 Kommunizieren und Kooperieren 7, 9 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (10) L BTV Formen von Vorurteilen, Stereotypen, Klischees; Minderheitenschutz; Personale und gesellschaftliche Vielfalt; Wertorientiertes Handeln L PG Sicherheit und Unfallschutz

(9) das Konzept der *asymmetrischen Verschlüsselung (privater/öffentlicher Schlüssel)* erklären

F IMP 3.3.1.4 Informationsgesellschaft und Datensicherheit (1)

L MB Informationelle Selbstbestimmung und Datenschutz; Informationstechnische Grundlagen; Kommunikation und Kooperation

* Falls das Fach in den Klassen 9 und 10 einstündig unterrichtet wird

3.2 Klassen 11/12 (Basisfach)

3.2.1 Daten und Codierung

3.2.1.1 Bitebene

Die Schülerinnen und Schüler erweitern ihre Kenntnisse über Codierungen von Zahlen und Texten. Sie lernen vorzeichenbehaftete Zahlendarstellungen kennen und reflektieren die beim Rechnen mit endlicher Stellenzahl bzw. eingeschränktem Wertebereich auftretenden Besonderheiten.

Die Schülerinnen und Schüler lernen mit dem Huffmanverfahren ein weiteres verlustfreies Kompressionsverfahren kennen. Sie reflektieren dieses im Hinblick auf seine Eignung für verschiedene Einsatzszenarien.

Die Schülerinnen und Schüler können

(1) negative Zahlen in ihrer <i>Einerkomplement-</i> und <i>Zweierkomplementdarstellung</i> angeben und interpretieren
<p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>P 2.4 Analysieren und Bewerten 3</p>
(2) Addition und Subtraktion von Binärzahlen in <i>Zweierkomplementdarstellung</i> schriftlich durchführen und das Ergebnis interpretieren
<p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>P 2.4 Analysieren und Bewerten 3</p>
(3) Einschränkungen beim Rechnen mit endlicher Stellenzahl erläutern (zum Beispiel Overflow, Genauigkeit)
P 2.4 Analysieren und Bewerten 6
(4) [nur LF]
(5) [nur LF]
(6) Merkmale von Codes (unter anderem Umkehrbarkeit, Präfixfreiheit, feste/variable Bitlänge) erläutern
P 2.2 Modellieren und Implementieren 4
(7) das <i>Huffmanverfahren</i> als Beispiel für ein <i>verlustfreies Datenkompressionsverfahren</i> erläutern und die Codierung durch Erzeugung eines <i>Huffmanbaums</i> sowie Decodierung von Hand durchführen
<p>P 2.2 Modellieren und Implementieren 5</p> <p>I 3.2.1.2 Datenstrukturen (3)</p>
(8) [nur LF]
(9) [nur LF]
L MB Informationstechnische Grundlagen

3.2.1.2 Datenstrukturen

Ausgehend von Beispielen aus dem Alltag lernen die Schülerinnen und Schüler unterschiedliche Ansätze kennen, Daten mithilfe von Arrays, Listen, Bäumen und Graphen systematisch zu strukturieren. Je nach Codierungsform einer Datenstruktur existieren unterschiedliche Möglichkeiten, auf dieser zu navigieren, Elemente hinzuzufügen oder zu entfernen. Diese Möglichkeiten werden dann von Algorithmen auf Datenstrukturen zur Lösung komplexerer Problemstellungen genutzt.

Abstrakte Datentypen (ADT) kapseln ihre interne Datenstruktur und sind damit unabhängig von einer konkreten Implementierung. Die Schülerinnen und Schüler lernen mit dem Stack einen ADT kennen, der unter anderem zur Auflösung von Rekursion und Realisierung von Unterprogrammaufrufen Anwendung findet. Der ADT Queue eignet sich insbesondere, um Alltagssituationen zu modellieren, in denen Warteschlangen auftreten.

Die Schülerinnen und Schüler können

Datenstrukturen	
(1)	[nur LF]
(2)	<i>einfach verkettete Listen</i> als Beispiel einer linearen Struktur und Grundoperationen (zum Beispiel Einfügen, Löschen) implementieren
	<p>P 2.2 Modellieren und Implementieren 9</p> <p>I 3.2.2.2 Algorithmen auf Datenstrukturen (1)</p>
(3)	<i>Binärbäume</i> als Beispiel einer Baumstruktur implementieren
	<p>P 2.2 Modellieren und Implementieren 9</p> <p>I 3.2.2.2 Algorithmen auf Datenstrukturen (4)</p>
(4)	<i>Graphen</i> in den Repräsentationsformen <i>Adjazenzmatrix</i> und <i>Adjazenzliste</i> beschreiben
	P 2.3 Kommunizieren und Kooperieren 1
(5)	Begriffe aus der Graphentheorie (unter anderem <i>Knoten</i> , <i>Kanten</i> , <i>Knotengrad</i> , <i>Kreis/Zyklus</i>) und Eigenschaften von <i>Graphen</i> (unter anderem <i>gerichtet/ungerichtet</i> , <i>gewichtet/ungewichtet</i> , <i>zyklisch/azyklisch</i>) verwenden
	P 2.3 Kommunizieren und Kooperieren 1
(6)	[nur LF]
Abstrakte Datentypen	
(7)	[nur LF]
(8)	den <i>Abstrakten Datentyp (ADT) Stack</i> beschreiben (LIFO-Prinzip)
	P 2.4 Analysieren und Bewerten 3
(9)	den <i>Abstrakten Datentyp (ADT) Queue</i> beschreiben (FIFO-Prinzip)
	P 2.4 Analysieren und Bewerten 3
(10)	[nur LF]
(11)	[nur LF]
	L MB Informationstechnische Grundlagen

3.2.1.3 Relationale Datenbanksysteme

Relationale Datenbanksysteme ermöglichen die Organisation und Speicherung großer Datenmengen auf Basis des relationalen Datenmodells, bei dem Entitäten und ihre Beziehungen in Tabellen abgebildet werden. Die Schülerinnen und Schüler nutzen geeignete Modellierungssprachen, um relationale Datenbankschemata zu erstellen und setzen ein Datenbankverwaltungsprogramm ein, um konkrete Anwendungsfälle zu realisieren. Die Schülerinnen und Schüler setzen die Datenbanksprache SQL (Structured Query Language) ein, um über eine logische Abstraktionsschicht auf Datenbanken zuzugreifen. Sie formulieren Anfragen, um Informationen zu ermitteln.

Die Schülerinnen und Schüler können

(1) Komponenten (unter anderem <i>Datenbank</i> , <i>Datenbankschnittstelle</i> , <i>Datenbankverwaltungssystem</i>) und deren Funktionen von <i>Datenbanksystemen</i> beschreiben
(2) das relationale Datenbankmodell und wesentliche Begriffe (<i>Entität</i> , <i>Entitätstyp</i> , <i>Attribut</i> , <i>Relation</i> , <i>Kardinalität</i> , <i>Tabelle</i> , <i>Datensatz</i> , <i>Datenfeld</i>) beschreiben
(3) die Begriffe <i>Schlüssel</i> , <i>Primärschlüssel</i> und <i>Fremdschlüssel</i> erläutern
(4) Ausschnitte der Realität in einem Diagramm (Entity-Relationship-Diagramm oder UML-Klassendiagramm) modellieren
<p>P 2.1 Strukturieren und Vernetzen 2, 8</p> <p>P 2.2 Modellieren und Implementieren 5, 7, 13</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>I 3.2.2.1 Strukturierte und objektorientierte Programmierung (20)</p>
(5) ein <i>relationales Datenbankschema</i> in 3. Normalform erstellen, um unerwünschte <i>Redundanzen</i> zu vermeiden und damit das Risiko für Inkonsistenzen zu minimieren
<p>P 2.1 Strukturieren und Vernetzen 8</p> <p>P 2.2 Modellieren und Implementieren 5, 7</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p>
(6) Diagramme und <i>relationale Datenbankschemata</i> ineinander überführen
<p>P 2.1 Strukturieren und Vernetzen 8</p> <p>P 2.2 Modellieren und Implementieren 5, 7</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p>
(7) ein <i>Datenbankverwaltungsprogramm</i> zur Erstellung und Administration von <i>Datenbanken</i> nutzen
(8) Abfragen auf <i>Datenbanken</i> in der <i>Datenbanksprache SQL</i> (Projektion, Selektion und Verbund über WHERE) auch über mehrere Tabellen durchführen
<p>P 2.2 Modellieren und Implementieren 3, 9, 13</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>P 2.4 Analysieren und Bewerten 4</p>
(9) Manipulationen auf <i>Datenbanken</i> in der <i>Datenbanksprache SQL</i> (unter anderem Ändern, Einfügen und Löschen) durchführen
<p>P 2.1 Strukturieren und Vernetzen 5, 8</p> <p>P 2.2 Modellieren und Implementieren 9</p>
L MB Information und Wissen; Informationstechnische Grundlagen

3.2.2 Algorithmen

3.2.2.1 Strukturierte und objektorientierte Programmierung

Die Schülerinnen und Schüler erweitern ihre vorhandenen Programmierkenntnisse. Der bisher eher propädeutische Umgang mit unterschiedlichen Datentypen wird mit den vertieften Kenntnissen über Text- und Zahlcodierungen in Zusammenhang gebracht. Die Schülerinnen und Schüler setzen Datentypen zielgerichtet ein und beachten die Problematik bei Typumwandlungen.

Mit der *Objektorientierten Programmierung* (OOP) lernen die Schülerinnen und Schüler ein neues Programmierparadigma kennen. Sie implementieren eigene Klassen und regeln den Zugriff auf Attribute mithilfe von Zugriffsmethoden (Datenkapselung/Geheimnisprinzip). Sie nutzen Assoziation und Vererbung zur Modellierung und Implementierung von Klassenbeziehungen.

Bei der *Objektorientierten Modellierung* (OOM) entwerfen die Schülerinnen und Schüler Klassendiagramme in der Modellierungssprache UML (Unified Modelling Language).

Die Schülerinnen und Schüler können

Variablen und Datentypen	
(1)	die Begriffe <i>Variable</i> , <i>Bezeichner</i> , <i>Datentyp</i> , <i>Wert</i> , <i>Deklaration</i> , <i>Initialisierung</i> , <i>Wertzuzuweisung</i> und <i>Gültigkeitsbereich</i> erläutern
	P 2.1 Strukturieren und Vernetzen 2
(2)	<i>primitive Datentypen</i> für ganzzahlige Werte, Gleitkommawerte und boolesche Werte unter Beachtung ihres <i>Wertebereichs</i> verwenden
	P 2.1 Strukturieren und Vernetzen 2 I 3.2.1.1 Bitebene (3)
(3)	implizite und explizite <i>Typumwandlungen (type casts)</i> zur Konvertierung von Werten unterschiedlicher <i>Datentypen</i> ineinander verwenden und dabei auftretende Probleme beschreiben
	I 3.2.1.1 Bitebene (3)
(4)	Operationen auf <i>Zeichenketten</i> durchführen (unter anderem Verketteten)
Objektorientierung	
(5)	eigene <i>Klassendefinitionen</i> mit <i>Attributen</i> und <i>Methoden</i> implementieren
	P 2.1 Strukturieren und Vernetzen 2, 7 P 2.3 Kommunizieren und Kooperieren 1, 3
(6)	<i>Referenzvariablen/Referenztypen</i> mit <i>primitiven Variablen / primitiven Datentypen</i> vergleichen
(7)	[nur LF]
(8)	<i>Methoden</i> mit und ohne <i>Rückgabewert</i> sowie mit und ohne <i>Parameter</i> implementieren und den Begriff der <i>Methodensignatur</i> erläutern
	P 2.1 Strukturieren und Vernetzen 2, 7 P 2.2 Modellieren und Implementieren 9 P 2.3 Kommunizieren und Kooperieren 1, 3, 4
(9)	die Funktion von <i>Zugriffsmodifikatoren (public, private, protected)</i> erläutern und diese verwenden
(10)	die <i>Kapselung von Attributen und Methoden (Geheimnisprinzip)</i> erläutern und in eigenen Implementierungen verwenden (unter anderem Zugriff auf Attribute über <i>Zugriffsmethoden</i>)
	P 2.1 Strukturieren und Vernetzen 7
(11)	[nur LF]
(12)	[nur LF]
(13)	die Rolle von <i>Konstruktoren</i> beim Erzeugen von <i>Instanzen einer Klasse (Objekten)</i> erläutern und diese implementieren
	P 2.1 Strukturieren und Vernetzen 7

(14) [nur LF]
(15) [nur LF]
(16) das Konzept der <i>Vererbung</i> erläutern und in Modellierungen und Implementierungen anwenden
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 3 P 2.2 Modellieren und Implementieren 4, 9
(17) [nur LF]
(18) [nur LF]
Modellierung und Dokumentation
(19) <i>Algorithmen</i> in den Notationsformen <i>Nassi-Shneiderman-Diagramm</i> und <i>Pseudocode</i> darstellen und interpretieren
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 5, 8 P 2.2 Modellieren und Implementieren 5, 7 P 2.3 Kommunizieren und Kooperieren 1, 2, 3, 4
(20) <i>UML-Klassendiagramme</i> mit <i>Vererbung</i> und <i>Assoziation</i> (auch rekursiv) entwerfen und zur Modellierung nutzen
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 3, 7, 8 P 2.2 Modellieren und Implementieren 1, 4, 5, 7 P 2.3 Kommunizieren und Kooperieren 1 I 3.2.1.3 Relationale Datenbanksysteme (4)
(21) ihren Programmcode nach gängigen Standards kommentieren und dazu vorhandene Hilfsmittel der Entwicklungsumgebung nutzen
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 2 P 2.3 Kommunizieren und Kooperieren 1, 3 L MB Kommunikation und Kooperation L PG Selbstregulation und Lernen
(22) Dokumentationen zu gegebenem Quellcode und Bibliotheken nutzen
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 1, 6, 10 P 2.3 Kommunizieren und Kooperieren 4 I 3.1.2 Algorithmen (9) L MB Kommunikation und Kooperation L PG Selbstregulation und Lernen
<ul style="list-style-type: none"> L MB Informationstechnische Grundlagen

3.2.2.2 Algorithmen auf Datenstrukturen


















Algorithmen zur Verarbeitung größerer Datenmengen setzen voraus, dass die Daten in einer geeigneten Struktur vorliegen. Für viele Algorithmen spielen dabei die Grundprobleme Suchen, Traversieren und Sortieren eine Rolle.

Suchverfahren werden benutzt, um die Existenz und die Position eines Element in einer Datenstruktur zu ermitteln (zum Beispiel bei der Artikelsuche in einem Webshop, bei der Rufnummernsuche im Telefonbuch, bei der Webrecherche). Die Schülerinnen und Schüler verwenden abhängig von der jeweiligen Problemstellung unterschiedliche Suchverfahren.

Sortierverfahren spielen in der Informatik eine große Rolle, da viele Algorithmen effizienter arbeiten, falls die Daten sortiert vorliegen. Dies lässt sich mithilfe von einfachen Sortierverfahren wie Selectionsort oder höheren Sortierverfahren wie Quicksort lösen.

Für ein und dasselbe Problem gibt es oftmals verschiedene Lösungsverfahren, die sich in Speicher- und Zeitkomplexität sowie – bei Näherungsverfahren – in der Genauigkeit der Lösung unterscheiden.

Die Schülerinnen und Schüler können

(1) die <i>lineare Suche</i> auf <i>Array</i> und <i>Liste</i> implementieren
<ul style="list-style-type: none">  2.2 Modellieren und Implementieren 9  3.1.2 Algorithmen (12), (13)  3.2.1.2 Datenstrukturen (2)
(2) die <i>binäre Suche</i> auf sortierten <i>Arrays</i> implementieren
<ul style="list-style-type: none">  2.2 Modellieren und Implementieren 9  3.1.2 Algorithmen (12), (13)
(3) [nur LF]
(4) <i>Breitensuche</i> und <i>Tiefensuche</i> auf Bäumen beschreiben
<ul style="list-style-type: none">  2.2 Modellieren und Implementieren 9  2.4 Analysieren und Bewerten 3  3.2.1.2 Datenstrukturen (3), (5)  3.2.2.3 Rekursion (1)
(5) elementare vergleichsbasierte Sortierverfahren (<i>Bubblesort</i> , <i>Selectionsort</i> , <i>Insertionsort</i>) beschreiben, händisch durchführen und eines davon implementieren
<ul style="list-style-type: none">  2.2 Modellieren und Implementieren 9, 11  2.4 Analysieren und Bewerten 3
(6) ein höheres vergleichsbasiertes rekursives Sortierverfahren (zum Beispiel <i>Mergesort</i> , <i>Quicksort</i>) beschreiben
<ul style="list-style-type: none">  2.2 Modellieren und Implementieren 9, 11  2.4 Analysieren und Bewerten 3  3.2.2.3 Rekursion (1), (2), (3)
(7) [nur LF]
(8) Sortierverfahren hinsichtlich der Eigenschaften <i>Laufzeit</i> (in O-Notation), <i>Speicherbedarf</i> und <i>Stabilität</i> vergleichen und die Begriffe <i>best case</i> , <i>worst case</i> und <i>average case</i> erläutern
 2.4 Analysieren und Bewerten 3, 6
(9) [nur LF]
(10) einen Algorithmus zur Lösung des <i>Problems des kürzesten Pfades</i> (zum Beispiel <i>Dijkstra</i>) beschreiben und an Beispielen von Hand durchführen
 3.2.1.2 Datenstrukturen (5)
(11) [nur LF]
(12) [nur LF]
 MB Informationstechnische Grundlagen

3.2.2.3 Rekursion

Die Schülerinnen und Schüler ergänzen ihre Problemlösestrategien um rekursive Algorithmen und erkennen deren Relevanz, indem sie konkrete Problemstellungen (zum Beispiel Türme von Hanoi) lösen.

Die zahlreichen Teilgebiete, in denen rekursive Algorithmen Anwendung finden, zum Beispiel beim Traversieren, Suchen und Sortieren, machen ihnen die übergeordnete Bedeutung dieser Algorithmenklasse deutlich.

Die Schülerinnen und Schüler können

(1) zu geeigneten Problemstellungen (zum Beispiel Türme von Hanoi, Baumtraversierung) <i>rekursive Algorithmen</i> unter Angabe von <i>Rekursionsschritt</i> und <i>Rekursionsbasis</i> entwerfen und von Hand durchführen
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 3, 5, 6 P 2.2 Modellieren und Implementieren 1, 13 P 2.3 Kommunizieren und Kooperieren 1 I 3.2.2.2 Algorithmen auf Datenstrukturen (2), (4), (6)
(2) das <i>Divide-and-Conquer</i> -Prinzip an geeigneten Problemstellungen erläutern
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 6 P 2.2 Modellieren und Implementieren 4 I 3.2.2.2 Algorithmen auf Datenstrukturen (6)
(3) <i>rekursive Algorithmen</i> zu unterschiedlichen Problemstellungen (zum Beispiel Fakultätsfunktion, Fibonacci-Zahlen, Kochsche Schneeflocke) implementieren
P 2.2 Modellieren und Implementieren 9
(4) [nur LF]
(5) [nur LF]
(6) [nur LF]
L MB Informationstechnische Grundlagen

3.2.3 Rechner und Netze

Anhand des Von-Neumann-Rechners lernen die Schülerinnen und Schüler eine konkret realisierbare Maschine und ihren inneren Aufbau kennen. Er stellt andererseits eine von der konkreten Realisierung unabhängige Modellmaschine dar, bei der Daten und Programm im selben Speicher liegen und deren Funktionalität auf einem ständigen Befehlszyklus basiert. Sie können hardwareabhängige Mikroprogramme in einer Simulationsumgebung implementieren, um Makrobefehle einer Von-Neumann-Modellmaschine zu erstellen oder deren Makrobefehlssatz zu erweitern.

Die Schülerinnen und Schüler können

Rechner	
(1)	[nur LF]
(2)	[nur LF]
(3)	[nur LF]
(4)	[nur LF]
(5)	[nur LF]
(6)	[nur LF]
(7)	[nur LF]
(8)	Aufbau, Funktionsweise (unter anderem Befehlszyklus) und Komponenten (unter anderem <i>Adressbus, Datenbus, Rechenwerk, Register, Steuerwerk, Speicherwerk</i>) einer <i>Von-Neumann-Modellmaschine</i> beschreiben
	<p>P 2.1 Strukturieren und Vernetzen 6</p> <p>P 2.3 Kommunizieren und Kooperieren 2</p>
(9)	<i>Mikroprogramme</i> für Assemblerbefehle (zum Beispiel ADD, SUB, JMP) in einer Simulationsumgebung implementieren
	<p>P 2.1 Strukturieren und Vernetzen 5, 8</p> <p>P 2.2 Modellieren und Implementieren 9, 13</p>
(10)	[nur LF]
Netze	
(11)	[nur LF]
(12)	[nur LF]
(13)	[nur LF]
	L MB Informationstechnische Grundlagen

3.2.4 Informationsgesellschaft und Datensicherheit

Schülerinnen und Schüler nutzen im Alltag Technologien (zum Beispiel Identifikations- und Bezahlsysteme, Mehrfaktorauthentifizierung, Signaturen), denen kryptologische Systeme zugrunde liegen. Sie lernen den prinzipiellen Unterschied zwischen symmetrischen und asymmetrischen Verfahren kennen und erläutern, wie Infrastrukturen zur verschlüsselten Kommunikation genutzt werden.

Die Schülerinnen und Schüler können

(1) <i>asymmetrische</i> und <i>symmetrische Verschlüsselung</i> vergleichen (Schlüsselverwaltung, Schlüsseltausch, Geschwindigkeit)
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 1 P 2.4 Analysieren und Bewerten 3
(2) erklären, wie Nachrichten mit <i>asymmetrischer Verschlüsselung signiert</i> werden können
P 2.3 Kommunizieren und Kooperieren 2
(3) erläutern, dass die Sicherheit <i>asymmetrischer Kryptoverfahren auf Einwegfunktionen</i> basiert, für die ausschließlich Angriffe mit exponentieller Laufzeit bekannt sind
(4) [nur LF]
(5) Kryptographische Ziele (Geheimhaltung, Authentifizierung, Integritätssicherung) erläutern
P 2.3 Kommunizieren und Kooperieren 2
(6) Anwendungsbereiche von <i>Public-Key-Infrastrukturen</i> beschreiben (zum Beispiel digitaler Personalausweis, Transportverschlüsselung, Mailverschlüsselung) und im Hinblick auf Sicherheitsaspekte bewerten
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 9, 10 P 2.4 Analysieren und Bewerten 5, 8
(7) [nur LF]
(8) [nur LF]
(9) Maßnahmen zur Gewährleistung von <i>Datensicherheit</i> und <i>Datenschutz</i> beim Einsatz von Informatiksystemen erläutern
<ul style="list-style-type: none"> P 2.4 Analysieren und Bewerten 1, 3, 4, 7, 8, 9 L BTV Minderheitenschutz; Wertorientiertes Handeln
(10) Szenarien bewerten, in denen Daten massenweise erhoben, gespeichert und weiterverarbeitet werden
<ul style="list-style-type: none"> P 2.4 Analysieren und Bewerten 5, 8, 9 I 3.2.1.3 Relationale Datenbanksysteme L BTV Minderheitenschutz; Wertorientiertes Handeln L BNE Teilhabe, Mitwirkung, Mitbestimmung; Werte und Normen in Entscheidungssituationen L MB Information und Wissen; Informationelle Selbstbestimmung und Datenschutz; Informationstechnische Grundlagen; Kommunikation und Kooperation; Mediengesellschaft L PG Sicherheit und Unfallschutz L VB Verbraucherrechte

3.2.5 Automaten und formale Sprachen

Im Alltag begegnen den Schülerinnen und Schülern Phänomene, die sich mithilfe von Automatenmodellen erklären lassen (zum Beispiel Ampelschaltung, Grafische Benutzerschnittstelle). Sie nutzen die Darstellungsformen Zustandsdiagramm und Zustandsübergangstabelle, um Automaten zu modellieren. Bei der Modellierung realer Automaten (Leergutautomaten, Snackautomaten und Ticketautomaten) werden im Unterricht auch Automaten mit Ausgaben behandelt.

Die Schülerinnen und Schüler können

(1) Anwendungsbereiche beschreiben, in denen Automatenmodelle eingesetzt werden
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 4, 5 P 2.3 Kommunizieren und Kooperieren 8 P 2.4 Analysieren und Bewerten 1 L MB Information und Wissen; Produktion und Präsentation
(2) [nur LF]
(3) [nur LF]
(4) [nur LF]
(5) [nur LF]
(6) [nur LF]
(7) einen <i>endlichen Automaten</i> mit und ohne Ausgabe (unter anderem <i>Mealy-Automat</i> als <i>Zustandsdiagramm</i>) zur Modellierung von Abläufen verwenden
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1
(8) zu einem <i>endlichen Automaten</i> die Darstellungsformen <i>Zustandsdiagramm</i> und <i>Zustandsübergangstabelle</i> ineinander überführen
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1
(9) [nur LF]
(10) [nur LF]
(11) [nur LF]
(12) [nur LF]
(13) [nur LF]
(14) [nur LF]
(15) Einsatzbereiche für <i>endliche Automaten</i> (zum Beispiel Substringproblem, GUI, zustandsbasierte Protokolle) erläutern
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 8 F D 3.3.2.1 Struktur von Äußerungen F D 3.4.2.1 Struktur von Äußerungen F D 3.5.2.1 Struktur von Äußerungen L MB Informationstechnische Grundlagen

3.2.6 Projektmanagement

Komplexere Projekte lassen sich heute nur noch in Teams planen und realisieren. Zur Planung und Durchführung haben sich verschiedene Methoden etabliert. Die Schülerinnen und Schüler lernen exemplarisch ein Vorgehensmodell und gängige Begriffe des Projektmanagements kennen.

Bei der Realisierung eigener Projekte mit dem Ziel, funktionsfähige Prototypen zu erstellen, wenden sie Methoden des Projektmanagements von der Planung über die Durchführung bis zur Dokumentation an.

Dabei vertiefen sie die erworbenen inhaltsbezogenen und prozessbezogenen Kompetenzen und vernetzen Kompetenzen aus mehreren Inhaltsbereichen miteinander.

So könnte ein Softwareprojekt eine Datenbank-Schnittstelle nutzen, um Anwendungs- und Nutzerdaten zu speichern, mithilfe eines Automatenmodells die Logik einer GUI modellieren oder kryptographische Verfahren zur sicheren Übertragung einsetzen.

Die Schülerinnen und Schüler können

(1) Aspekte des Projektmanagements (unter anderem <i>Projekt, Projektauftrag, Projektphasen, Anforderungen, Abnahme</i>) und ein Vorgehensmodell (zum Beispiel agiles Vorgehen, iterativ, Wasserfallmodell) erklären
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 5, 6, 7, 8, 9 P 2.3 Kommunizieren und Kooperieren 5 P 2.4 Analysieren und Bewerten 7
(2) [nur LF]
(3) [nur LF]
(4) einen Projektentwurf erstellen und Anforderungen an das <i>Projekt</i> in geeigneter Form (zum Beispiel Lastenheft, Pflichtenheft, User Stories, Systemfunktionen) beschreiben, ein (Teil-)Projekt durchführen und einen Prototyp erstellen
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 5, 6, 7, 8, 9 P 2.2 Modellieren und Implementieren 2, 3, 4, 5, 8, 9 P 2.3 Kommunizieren und Kooperieren 2, 5, 6, 7, 9, 10 P 2.4 Analysieren und Bewerten 3, 4, 7
<ul style="list-style-type: none"> L BNE Demokratiefähigkeit; Teilhabe, Mitwirkung, Mitbestimmung L BO Einschätzung und Überprüfung eigener Fähigkeiten und Potenziale; Fachspezifische und handlungsorientierte Zugänge zur Arbeits- und Berufswelt; Geschlechtsspezifische Aspekte bei der Berufswahl, Familien- und Lebensplanung; Kompetenzanalyse, Eignungstests und Entscheidungstrainings L BTV Konfliktbewältigung und Interessenausgleich; Selbstfindung und Akzeptanz anderer Lebensformen; Wertorientiertes Handeln L MB Information und Wissen; Kommunikation und Kooperation; Produktion und Präsentation L PG Selbstregulation und Lernen; Sicherheit und Unfallschutz L VB Qualität der Konsumgüter; Umgang mit eigenen Ressourcen

3.3 Klassen 11/12 (Leistungsfach)

3.3.1 Daten und Codierung

3.3.1.1 Bitebene

Die Schülerinnen und Schüler erweitern ihre Kenntnisse über Codierungen von Zahlen und Texten. Sie lernen nichtganze und vorzeichenbehaftete Zahlendarstellungen kennen und reflektieren die beim Rechnen mit endlicher Stellenzahl bzw. eingeschränktem Wertebereich auftretenden Besonderheiten.

Die Schülerinnen und Schüler lernen weitere auf unterschiedlichen Ansätzen basierende verlustfreie Kompressionsverfahren kennen. Sie reflektieren diese im Hinblick auf ihre Eignung für verschiedene Einsatzszenarien. Mit Hashfunktionen lernen die Schülerinnen und Schüler ein Hilfsmittel kennen, dass in vielen Teilbereichen der Informatik, unter anderem Datenbanksystemen und Kryptologie, eingesetzt wird.

Die Schülerinnen und Schüler können

(1) negative Zahlen in ihrer <i>Einerkomplement-</i> und <i>Zweierkomplementdarstellung</i> angeben und interpretieren
<p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>P 2.4 Analysieren und Bewerten 3</p>
(2) Addition und Subtraktion von Binärzahlen in <i>Zweierkomplementdarstellung</i> schriftlich durchführen und das Ergebnis interpretieren
<p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>P 2.4 Analysieren und Bewerten 3</p>
(3) Einschränkungen beim Rechnen mit endlicher Stellenzahl erläutern (zum Beispiel Overflow, Genauigkeit)
P 2.4 Analysieren und Bewerten 6
(4) die <i>Festkommadarstellung</i> als Beispiel für eine Codierung nichtganzer Zahlen erläutern
P 2.3 Kommunizieren und Kooperieren 1
(5) am Beispiel von Textcodierungen Erweiterungsmöglichkeiten von Codierungen beschreiben (zum Beispiel Biterweiterung, Escapezeichen, Codepages)
<p>P 2.2 Modellieren und Implementieren 4</p> <p>I 3.3.5 Automaten und formale Sprachen</p>
(6) Merkmale von Codes (unter anderem Umkehrbarkeit, Präfixfreiheit, feste/variable Bitlänge) erläutern
P 2.2 Modellieren und Implementieren 4
(7) das <i>Huffmanverfahren</i> als Beispiel für ein <i>verlustfreies Datenkompressionsverfahren</i> erläutern und die Codierung durch Erzeugung eines <i>Huffmanbaums</i> sowie Decodierung von Hand durchführen
<p>P 2.2 Modellieren und Implementieren 5</p> <p>I 3.3.1.2 Datenstrukturen (3)</p>
(8) das <i>LZW-Verfahren</i> als Beispiel für ein <i>Wörterbuchverfahren</i> erläutern und von Hand durchführen
(9) das Konzept von <i>Hashfunktionen</i> und Anforderungen an diese (unter anderem Nichtumkehrbarkeit, Vermeidung von <i>Kollisionen</i>) beschreiben und dieses an Beispielen (zum Beispiel Fingerprint, Integritätsprüfung, ISBN-Prüfsumme) erläutern
I 3.3.4 Informationsgesellschaft und Datensicherheit (4), (5)
L MB Informationstechnische Grundlagen

3.3.1.2 Datenstrukturen

Ausgehend von Beispielen aus dem Alltag lernen die Schülerinnen und Schüler unterschiedliche Ansätze kennen, Daten mithilfe von Arrays, Listen, Bäumen und Graphen systematisch zu strukturieren. Je nach Codierungsform einer Datenstruktur existieren unterschiedliche Möglichkeiten, auf dieser zu navigieren, Elemente hinzuzufügen oder zu entfernen. Diese Möglichkeiten werden dann von Algorithmen auf Datenstrukturen zur Lösung komplexerer Problemstellungen genutzt.

Abstrakte Datentypen (ADT) kapseln ihre interne Datenstruktur und sind damit unabhängig von einer konkreten Implementierung. Die Schülerinnen und Schüler lernen mit dem Stack einen ADT kennen, der unter anderem zur Auflösung von Rekursion und Realisierung von Unterprogrammaufrufen Anwendung findet. Der ADT Queue eignet sich insbesondere, um Alltagssituationen zu modellieren, in denen Warteschlangen auftreten.

Die Schülerinnen und Schüler können

(1) zweidimensionale <i>Arrays</i> in ihrer Implementierung verwenden
<p>P 2.2 Modellieren und Implementieren 9</p> <p>I 3.3.2.3 Rekursion (6)</p>
(2) <i>einfach verkettete Listen</i> als Beispiel einer linearen Struktur und Grundoperationen (zum Beispiel Einfügen, Löschen) implementieren
<p>P 2.2 Modellieren und Implementieren 9</p> <p>I 3.3.2.2 Algorithmen auf Datenstrukturen (1)</p>
(3) <i>Binärbäume</i> als Beispiel einer Baumstruktur implementieren
<p>P 2.2 Modellieren und Implementieren 9</p> <p>I 3.3.2.2 Algorithmen auf Datenstrukturen (3), (4)</p>
(4) <i>Graphen</i> in den Repräsentationsformen <i>Adjazenzmatrix</i> und <i>Adjazenzliste</i> beschreiben
P 2.3 Kommunizieren und Kooperieren 1
(5) Begriffe aus der Graphentheorie (unter anderem <i>Knoten</i> , <i>Kanten</i> , <i>Knotengrad</i> , <i>Kreis/Zyklus</i>) und Eigenschaften von <i>Graphen</i> (unter anderem <i>gerichtet/ungerichtet</i> , <i>gewichtet/ungewichtet</i> , <i>zyklisch/azyklisch</i>) verwenden
I 3.3.2.2 Algorithmen auf Datenstrukturen (9), (10)
(6) erläutern, dass <i>Bäume</i> spezielle <i>Graphen</i> sind (Begriffe: <i>Wurzel</i> , <i>innerer Knoten</i> , <i>Blatt</i>)
<p>P 2.3 Kommunizieren und Kooperieren 2</p> <p>I 3.3.2.2 Algorithmen auf Datenstrukturen (9), (10)</p>
(7) das Konzept des <i>Abstrakten Datentyps</i> (unter anderem anhand von <i>Set</i>) erläutern
<p>P 2.3 Kommunizieren und Kooperieren 2</p> <p>I 3.3.2.1 Strukturierte und objektorientierte Programmierung (15)</p>
(8) den <i>Abstrakten Datentyp (ADT) Stack</i> mit den Operationen <i>isEmpty</i> , <i>push</i> , <i>pop</i> und <i>top</i> beschreiben (LIFO-Prinzip) und mithilfe einer geeigneten Datenstruktur implementieren
<p>P 2.4 Analysieren und Bewerten 3</p> <p>I 3.3.2.1 Strukturierte und objektorientierte Programmierung (15)</p>
(9) den <i>Abstrakten Datentyp (ADT) Queue</i> mit den Operationen <i>isEmpty</i> , <i>enqueue</i> , <i>dequeue</i> und <i>front</i> beschreiben (FIFO-Prinzip) und mithilfe einer geeigneten Datenstruktur implementieren
<p>P 2.4 Analysieren und Bewerten 3</p> <p>I 3.3.2.1 Strukturierte und objektorientierte Programmierung (15)</p>
(10) Datenstrukturen zur Modellierung und Lösung ausgewählter Anwendungsfälle (zum Beispiel Straßennetz, Geldflüsse, Freundschaftsbeziehungen in sozialen Netzwerken, Organigramm, To-do-Liste, Klammerausdrücke, Rangierbahnhof) nutzen
P 2.4 Analysieren und Bewerten 2
(11) einen Algorithmus auf Graphen implementieren (unter Verwendung geeigneter Bibliotheken oder Frameworks)
<p>P 2.2 Modellieren und Implementieren 6, 9</p> <p>P 2.3 Kommunizieren und Kooperieren 4</p>
L MB Informationstechnische Grundlagen

3.3.1.3 Relationale Datenbanksysteme

Relationale Datenbanksysteme ermöglichen die Organisation und Speicherung großer Datenmengen auf Basis des relationalen Datenmodells, bei dem Entitäten und ihre Beziehungen in Tabellen abgebildet werden. Die Schülerinnen und Schüler nutzen geeignete Modellierungssprachen, um relationale Datenbankschemata zu erstellen und setzen ein Datenbankverwaltungsprogramm ein, um konkrete Anwendungsfälle zu realisieren. Die Schülerinnen und Schüler setzen die Datenbanksprache SQL (Structured Query Language) ein, um über eine logische Abstraktionsschicht auf Datenbanken zuzugreifen. Sie formulieren Anfragen, um Informationen zu ermitteln, Daten einzufügen, zu löschen oder zu manipulieren. Die Schülerinnen und Schüler können

(1) Komponenten (unter anderem <i>Datenbank</i> , <i>Datenbankschnittstelle</i> , <i>Datenbankverwaltungssystem</i>) und deren Funktionen von <i>Datenbanksystemen</i> beschreiben
(2) das relationale Datenbankmodell und wesentliche Begriffe (<i>Entität</i> , <i>Entitätstyp</i> , <i>Attribut</i> , <i>Relation</i> , <i>Kardinalität</i> , <i>Tabelle</i> , <i>Datensatz</i> , <i>Datenfeld</i>) beschreiben
(3) die Begriffe <i>Schlüssel</i> , <i>Primärschlüssel</i> und <i>Fremdschlüssel</i> erläutern
(4) Ausschnitte der Realität in einem Diagramm (<i>Entity-Relationship-Diagramm</i> und <i>UML-Klassendiagramm</i>) modellieren
<p>P 2.1 Strukturieren und Vernetzen 2, 8</p> <p>P 2.2 Modellieren und Implementieren 5, 7, 13</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>I 3.3.2.1 Strukturierte und objektorientierte Programmierung (20)</p>
(5) ein <i>relationales Datenbankschema</i> in 3. Normalform erstellen, um unerwünschte <i>Redundanzen</i> zu vermeiden und damit das Risiko für Inkonsistenzen zu minimieren
<p>P 2.1 Strukturieren und Vernetzen 8</p> <p>P 2.2 Modellieren und Implementieren 5, 7</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p>
(6) Diagramme und <i>relationale Datenbankschemata</i> ineinander überführen
<p>P 2.1 Strukturieren und Vernetzen 8</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p>
(7) ein <i>Datenbankverwaltungsprogramm</i> zur Erstellung und Administration von <i>Datenbanken</i> nutzen
(8) Abfragen auf <i>Datenbanken</i> in der <i>Datenbanksprache SQL</i> (Aggregatfunktion, Gruppierung, Projektion, Selektion und Verbund über WHERE) auch über mehrere Tabellen durchführen
<p>P 2.2 Modellieren und Implementieren 3, 9, 13</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>P 2.4 Analysieren und Bewerten 4</p>
(9) Manipulationen auf <i>Datenbanken</i> in der <i>Datenbanksprache SQL</i> (unter anderem Ändern, Einfügen und Löschen) durchführen
<p>P 2.1 Strukturieren und Vernetzen 5, 8</p> <p>P 2.2 Modellieren und Implementieren 9</p>
L MB Information und Wissen; Informationstechnische Grundlagen

3.3.2 Algorithmen

3.3.2.1 Strukturierte und objektorientierte Programmierung

Die Schülerinnen und Schüler erweitern ihre vorhandenen Programmierkenntnisse. Der bisher eher propädeutische Umgang mit unterschiedlichen Datentypen wird mit den vertieften Kenntnissen über Text- und Zahlcodierungen in Zusammenhang gebracht. Die Schülerinnen und Schüler setzen Datentypen zielgerichtet ein und beachten die Problematik bei Typumwandlungen.

Mit der *Objektorientierten Programmierung* (OOP) lernen die Schülerinnen und Schüler ein neues Programmierparadigma kennen. Sie implementieren eigene Klassen und regeln den Zugriff auf Attribute mithilfe von Zugriffsmethoden (Datenkapselung/Geheimnisprinzip). Sie nutzen Assoziation und Vererbung zur Modellierung und Implementierung von Klassenbeziehungen.

Bei der *Objektorientierten Modellierung* (OOM) entwerfen die Schülerinnen und Schüler Klassendiagramme in der Modellierungssprache UML (Unified Modelling Language).

Die Schülerinnen und Schüler können

Variablen und Datentypen	
(1)	die Begriffe <i>Variable</i> , <i>Bezeichner</i> , <i>Datentyp</i> , <i>Wert</i> , <i>Deklaration</i> , <i>Initialisierung</i> , <i>Wertzuweisung</i> und <i>Gültigkeitsbereich</i> erläutern
	P 2.1 Strukturieren und Vernetzen 2
(2)	<i>primitive Datentypen</i> für ganzzahlige Werte, Gleitkommawerte und boolesche Werte unter Beachtung ihres <i>Wertebereichs</i> verwenden
	P 2.1 Strukturieren und Vernetzen 2 I 3.3.1.1 Bitebene (3)
(3)	implizite und explizite <i>Typumwandlungen</i> (<i>type casts</i>) zur Konvertierung von Werten unterschiedlicher <i>Datentypen</i> ineinander verwenden und dabei auftretende Probleme beschreiben
	I 3.3.1.1 Bitebene (3)
(4)	Operationen auf <i>Zeichenketten</i> (unter anderem Vergleichen, Verketteten, Zugriff auf einzelne Zeichen, Umwandlung in Zahlen) durchführen
Objektorientierung	
(5)	eigene <i>Klassendefinitionen</i> mit <i>Attributen</i> und <i>Methoden</i> implementieren
	P 2.1 Strukturieren und Vernetzen 2, 7 P 2.3 Kommunizieren und Kooperieren 1, 3
(6)	<i>Referenzvariablen/Referenztypen</i> mit <i>primitiven Variablen / primitiven Datentypen</i> vergleichen
(7)	mögliche Ursachen und Auswirkungen von Nullpointerfehlern erläutern
	P 2.2 Modellieren und Implementieren 12 P 2.4 Analysieren und Bewerten 3, 4
(8)	<i>Methoden</i> mit und ohne <i>Rückgabewert</i> sowie mit und ohne <i>Parameter</i> implementieren und den Begriff der <i>Methodensignatur</i> erläutern
	P 2.1 Strukturieren und Vernetzen 2, 7 P 2.2 Modellieren und Implementieren 9 P 2.3 Kommunizieren und Kooperieren 1, 3, 4
(9)	die Funktion von <i>Zugriffsmodifikatoren</i> (<i>public</i> , <i>private</i> , <i>protected</i>) erläutern und diese verwenden
(10)	die <i>Kapselung</i> von <i>Attributen</i> und <i>Methoden</i> (<i>Geheimnisprinzip</i>) erläutern und in eigenen Implementierungen verwenden (unter anderem Zugriff auf Attribute über <i>Zugriffsmethoden</i>)
	P 2.1 Strukturieren und Vernetzen 7
(11)	den Unterschied zwischen <i>Objektattributen/Objektmethoden</i> und <i>Klassenattributen/Klassenmethoden</i> erklären

(12) den Lebenszyklus von <i>Objekten</i> (unter anderem <i>Instanziierung</i> , Verwendung, Zerstörung) beschreiben
(13) die Rolle von <i>Konstruktoren</i> beim Erzeugen von <i>Instanzen einer Klasse (Objekten)</i> erläutern und diese implementieren
P 2.1 Strukturieren und Vernetzen 7
(14) generische Datentypen bei der <i>Instanziierung vorgegebener Klassen</i> verwenden
P 2.2 Modellieren und Implementieren 4
(15) sprachliche Mittel zur Sammlung gleichartiger <i>Objekte</i> (Collection) und zum Durchlaufen aller Elemente der Sammlung (zum Beispiel Iteration, foreach, Lambdaausdrücke etc.) nutzen
P 2.1 Strukturieren und Vernetzen 4
P 2.2 Modellieren und Implementieren 6, 10
P 2.3 Kommunizieren und Kooperieren 4
I 3.3.1.2 Datenstrukturen (7), (8), (9)
(16) das Konzept der <i>Vererbung</i> erläutern und in Modellierungen und Implementierungen anwenden
P 2.1 Strukturieren und Vernetzen 3
P 2.2 Modellieren und Implementieren 4, 9
(17) das Prinzip der <i>Polymorphie</i> erläutern und anwenden und dabei auch sichere <i>class casts</i> verwenden
(18) <i>abstrakte Klassen</i> und <i>abstrakte Methoden</i> nutzen
P 2.2 Modellieren und Implementieren 4, 9
P 2.3 Kommunizieren und Kooperieren 1
Modellierung und Dokumentation
(19) <i>Algorithmen</i> in den Notationsformen <i>Nassi-Shneiderman-Diagramm</i> und <i>Pseudocode</i> darstellen und interpretieren
P 2.1 Strukturieren und Vernetzen 5, 8
P 2.2 Modellieren und Implementieren 5, 7
P 2.3 Kommunizieren und Kooperieren 1, 2, 3, 4
(20) <i>UML-Klassendiagramme</i> mit <i>Vererbung</i> und <i>Assoziation</i> (auch rekursiv) entwerfen und zur Modellierung nutzen
P 2.1 Strukturieren und Vernetzen 3, 7, 8
P 2.2 Modellieren und Implementieren 1, 4, 5, 7
P 2.3 Kommunizieren und Kooperieren 1
I 3.3.1.3 Relationale Datenbanksysteme (4)
(21) ihren Programmcode nach gängigen Standards kommentieren und dazu vorhandene Hilfsmittel der Entwicklungsumgebung nutzen
P 2.2 Modellieren und Implementieren 2
P 2.3 Kommunizieren und Kooperieren 1, 3
L MB Kommunikation und Kooperation
L PG Selbstregulation und Lernen
(22) Dokumentationen zu gegebenem Quellcode und Bibliotheken nutzen
P 2.2 Modellieren und Implementieren 1, 6, 10
P 2.3 Kommunizieren und Kooperieren 4
I 3.1.2 Algorithmen (9)
L MB Kommunikation und Kooperation
L PG Selbstregulation und Lernen
L MB Informationstechnische Grundlagen

3.3.2.2 Algorithmen auf Datenstrukturen

Algorithmen zur Verarbeitung größerer Datenmengen setzen voraus, dass die Daten in einer geeigneten Struktur vorliegen. Für viele Algorithmen spielen dabei die Grundprobleme Suchen, Traversieren und Sortieren eine Rolle.

Suchverfahren werden benutzt, um die Existenz und die Position eines Elements in einer Datenstruktur zu ermitteln (zum Beispiel bei der Artikelsuche in einem Webshop, bei der Rufnummernsuche im Telefonbuch, bei der Webrecherche). Die Schülerinnen und Schüler verwenden abhängig von der jeweiligen Problemstellung und Datenstruktur (zum Beispiel Array, Baum) unterschiedliche Verfahren (lineare und binäre Suche, Breiten- und Tiefensuche).

Sortierverfahren spielen in der Informatik eine große Rolle, da viele Algorithmen effizienter arbeiten, falls die Daten sortiert vorliegen. Dies lässt sich mithilfe von einfachen Sortierverfahren wie Selectionsort oder höheren Sortierverfahren wie Quicksort lösen.

Für ein und dasselbe Problem gibt es oftmals verschiedene Lösungsverfahren, die sich in Speicher- und Zeitkomplexität sowie – bei Näherungsverfahren – in der Genauigkeit der Lösung unterscheiden. Neben klassischen Such- und Sortierproblemen werden auch komplexere Problemstellungen untersucht.

Die Schülerinnen und Schüler können

(1) die <i>lineare Suche</i> auf <i>Array</i> und <i>Liste</i> implementieren
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 9 I 3.1.2 Algorithmen (12), (13) I 3.3.1.2 Datenstrukturen (2)
(2) die <i>binäre Suche</i> auf sortierten <i>Arrays</i> implementieren
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 9 I 3.1.2 Algorithmen (12), (13)
(3) <i>Inorder</i> -, <i>Postorder</i> - und <i>Preorder</i> -Traversierung auf <i>Binärbäumen</i> händisch durchführen und Anwendungsbeispiele nennen
P 2.4 Analysieren und Bewerten 3
(4) <i>Breitensuche</i> und <i>Tiefensuche</i> auf <i>Bäumen</i> beschreiben und implementieren
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 9 P 2.4 Analysieren und Bewerten 3 I 3.3.1.2 Datenstrukturen (3), (5), (6) I 3.3.2.3 Rekursion (1), (6)
(5) elementare vergleichsbasierte Sortierverfahren (<i>Bubblesort</i> , <i>Selectionsort</i> , <i>Insertionsort</i>) beschreiben, händisch durchführen und implementieren
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 9, 11 P 2.4 Analysieren und Bewerten 3
(6) höhere vergleichsbasierte rekursive Sortierverfahren (<i>Mergesort</i> , <i>Quicksort</i>) beschreiben, händisch durchführen und eines davon implementieren
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 9, 11 P 2.4 Analysieren und Bewerten 3 I 3.3.2.3 Rekursion (1), (2), (4)
(7) ein nicht-vergleichsbasiertes Verfahren (zum Beispiel <i>Bucketsort</i> , <i>Countingsort</i> , <i>Radixsort</i> , ...) beschreiben
P 2.4 Analysieren und Bewerten 3
(8) Sortierverfahren hinsichtlich der Eigenschaften <i>Laufzeit</i> (in O-Notation), <i>Speicherbedarf</i> und <i>Stabilität</i> vergleichen und die Begriffe <i>best case</i> , <i>worst case</i> und <i>average case</i> erläutern
P 2.4 Analysieren und Bewerten 3, 6
(9) <i>Breitensuche</i> und <i>Tiefensuche</i> auf <i>Graphen</i> beschreiben, auf reale Problemstellungen anwenden und händisch durchführen
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 13 P 2.4 Analysieren und Bewerten 3, 6 I 3.3.1.2 Datenstrukturen (5), (6)

(10) einen <i>Algorithmus</i> (zum Beispiel Prim, Kruskal) zur Bestimmung eines <i>Minimum Spanning Tree</i> und einen zur Lösung des <i>Problems des kürzesten Pfades</i> (zum Beispiel Dijkstra, Bellman-Ford) beschreiben und an Beispielen von Hand durchführen
I 3.3.1.2 Datenstrukturen (5), (6) L BNE Komplexität und Dynamik nachhaltiger Entwicklung
(11) erläutern, dass nicht bekannt ist, ob für jedes Problem eine in Polynomialzeit berechenbare Lösung existiert, und können dafür Beispiele angeben
P 2.4 Analysieren und Bewerten 6
(12) Strategien (zum Beispiel Greedy) zur Bestimmung von Näherungslösungen in polynomieller Laufzeit beschreiben und an geeigneten Problemstellungen (zum Beispiel 4-Farben-Problem, Dominating Sets) von Hand durchführen
P 2.4 Analysieren und Bewerten 3, 6 L MB Informationstechnische Grundlagen

3.3.2.3 Rekursion

Die Schülerinnen und Schüler ergänzen ihre Problemlösestrategien um rekursive Algorithmen und erkennen deren Relevanz, indem sie konkrete Problemstellungen (zum Beispiel Türme von Hanoi, Acht-Damen-Problem) lösen.

Die zahlreichen Teilgebiete, in denen rekursive Algorithmen Anwendung finden, zum Beispiel beim Traversieren, Suchen und Sortieren, machen ihnen die übergeordnete Bedeutung dieser Algorithmenklasse deutlich. Mithilfe von Beispielen erarbeiten die Schülerinnen und Schüler Kriterien, anhand derer sich der sinnvolle Einsatz von rekursiven Algorithmen gegenüber iterativen Algorithmen bewerten lässt.

Sie verstehen, wie rekursive Algorithmen von Rechnern ausgeführt werden, indem sie Rekursionsabläufe darstellen.

Die Schülerinnen und Schüler können

(1) zu geeigneten Problemstellungen (zum Beispiel Türme von Hanoi, Baumtraversierung) <i>rekursive Algorithmen</i> unter Angabe von <i>Rekursionsschritt</i> und <i>Rekursionsbasis</i> entwerfen und von Hand durchführen
<p>P 2.1 Strukturieren und Vernetzen 3, 5, 6</p> <p>P 2.2 Modellieren und Implementieren 1, 13</p> <p>P 2.3 Kommunizieren und Kooperieren 1</p> <p>I 3.3.2.2 Algorithmen auf Datenstrukturen (2), (4), (6), (9)</p>
(2) das <i>Divide-and-Conquer</i> -Prinzip an geeigneten Problemstellungen erläutern
<p>P 2.1 Strukturieren und Vernetzen 6</p> <p>P 2.2 Modellieren und Implementieren 4</p> <p>I 3.3.2.2 Algorithmen auf Datenstrukturen (6)</p>
(3) <i>rekursive Algorithmen</i> zu unterschiedlichen Problemstellungen (zum Beispiel Fakultätsfunktion, Fibonacci-Zahlen, Kochsche Schneeflocke) implementieren
P 2.2 Modellieren und Implementieren 9
(4) Rekursionsabläufe darstellen (unter anderem am <i>call stack</i> , <i>Baum</i>)
<p>P 2.1 Strukturieren und Vernetzen 3</p> <p>P 2.3 Kommunizieren und Kooperieren 1, 2</p>
(5) <i>iterative Algorithmen</i> und <i>rekursive Algorithmen</i> zur Lösung derselben Problemstellung vergleichen (unter anderem hinsichtlich <i>Laufzeit</i>) und bewerten
P 2.4 Analysieren und Bewerten 3, 4
(6) das Prinzip des <i>Backtrackings</i> anhand einer geeigneten Problemstellung (zum Beispiel Acht-Damen-Problem, Magische Quadrate, Zyklensuche) erläutern
<p>I 3.3.1.2 Datenstrukturen (1)</p> <p>I 3.3.2.2 Algorithmen auf Datenstrukturen (4), (9)</p>
L MB Informationstechnische Grundlagen

3.3.3 Rechner und Netze

Die boolesche Algebra ist die theoretische Grundlage von Schaltnetzen und Schaltwerken, aus denen die Hardware real existierender Informatiksysteme besteht. Ausgehend von einfachen Wahrheitstabellen lernen die Schülerinnen und Schüler Methoden kennen, um zunehmend komplexere Schaltnetze zu entwickeln und zu optimieren.

Anhand des Von-Neumann-Rechners lernen die Schülerinnen und Schüler eine konkret realisierbare Maschine und ihren inneren Aufbau kennen. Er stellt andererseits eine von der konkreten Realisierung unabhängige Modellmaschine dar, bei der Daten und Programm im selben Speicher liegen und deren Funktionalität auf einem ständigen Befehlszyklus basiert. Sie können hardwareabhängige Mikroprogramme in einer Simulationsumgebung implementieren, um Makrobefehle einer Von-Neumann-Modellmaschine zu erstellen oder deren Makrobefehlssatz zu erweitern. Diesen nutzen sie exemplarisch, um einfache hardwareunabhängige Programme zu implementieren.

Mehrere miteinander verbundene Rechner können mithilfe von Protokollen, welche die Kommunikation regeln, kommunizieren. Die Schülerinnen und Schüler bauen ihre bisherigen Kenntnisse über Protokolle aus, indem sie das Prinzip der paketorientierten Datenübertragung und das vereinfachte Schichtenmodell an einem konkreten Beispiel erläutern. Sie stellen Kommunikationsabläufe mithilfe von Sequenzdiagrammen dar.

Die Schülerinnen und Schüler können

Rechner
(1) aus <i>logischen Gattern</i> (unter anderem <i>AND</i> , <i>OR</i> , <i>XOR</i> , <i>NOT</i>) Schaltnetze entwerfen, diese untersuchen und ihre <i>Wahrheitstabellen</i> ermitteln
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 6, 8 P 2.2 Modellieren und Implementieren 7, 13 P 2.3 Kommunizieren und Kooperieren 1 P 2.4 Analysieren und Bewerten 1
(2) Aufbau und Funktion von <i>Halbaddierer</i> und <i>Volladdierer</i> beschreiben und daraus in einer Simulationsumgebung einen Mehrbitaddierer erstellen
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 6, 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1 I 3.3.1.1 Bitebene (2), (3)
(3) Aufbau und Funktion eines bistabilen Bauteils (zum Beispiel Latch, Flipflop) als Beispiel für einen 1-Bit-Speicher beschreiben und in einer Simulationsumgebung <i>SR-Latch</i> und <i>D-Latch</i> erstellen
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 6, 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1
(4) Begriffe der <i>booleschen Algebra</i> (<i>boolescher Wert</i> , <i>Verknüpfungen</i> , <i>Basis</i>) erklären
(5) <i>Boolesche Terme</i> mithilfe der Rechengesetze (unter anderem <i>De Morgan'sche Regeln</i>) in eine vereinfachte Darstellung überführen
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 1, 2 P 2.4 Analysieren und Bewerten 3
(6) <i>Boolesche Funktionen</i> in <i>disjunktiver Normalform (DNF)</i> und <i>konjunktiver Normalform (KNF)</i> angeben, auch <i>kanonisch</i>
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 1
(7) <i>Boolesche Funktionen</i> in bis zu vier Variablen mithilfe von <i>KV-Diagrammen</i> in <i>Minimalform</i> überführen
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 1
(8) Aufbau, Funktionsweise (unter anderem <i>Befehlszyklus</i>) und Komponenten (unter anderem <i>Adressbus</i> , <i>Datenbus</i> , <i>Rechenwerk</i> , <i>Register</i> , <i>Steuerwerk</i> , <i>Speicherwerk</i>) einer <i>Von-Neumann-Modellmaschine</i> beschreiben
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 6 P 2.3 Kommunizieren und Kooperieren 2

(9) <i>Mikroprogramme</i> für Assemblerbefehle (zum Beispiel ADD, SUB, JMP) in einer Simulationsumgebung implementieren
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 5, 8 P 2.2 Modellieren und Implementieren 9, 13
(10) Assembler-Programme für eine <i>Von-Neumann-Modellmaschine</i> (zum Beispiel Multiplikation natürlicher Zahlen) in einer Simulationsumgebung implementieren
<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 5 P 2.2 Modellieren und Implementieren 9
Netze
(11) das Prinzip der <i>paketorientierten Übertragung</i> (unter anderem <i>Adressierung, Segmentierung</i>) erläutern
P 2.3 Kommunizieren und Kooperieren 2, 8
(12) das Zusammenwirken von <i>Protokollschichten</i> (unter anderem am Beispiel von <i>TCP/IP</i>) am vereinfachten Schichtenmodell (<i>Netzwerkschicht, Vermittlungsschicht, Transportschicht, Anwendungsschicht</i>) erläutern
<ul style="list-style-type: none"> P 2.3 Kommunizieren und Kooperieren 2, 8 L BNE Komplexität und Dynamik nachhaltiger Entwicklung; Kriterien für nachhaltigkeitsfördernde und -hemmende Handlungen
(13) den Kommunikationsablauf über ein <i>Protokoll</i> (unter anderem <i>TCP</i> mit Verbindungsaufbau und -abbau) in einem <i>Sequenzdiagramm</i> darstellen
<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 5 P 2.3 Kommunizieren und Kooperieren 1, 8 I 3.3.5 Automaten und formale Sprachen (1), (7), (15) L MB Kommunikation und Kooperation
L MB Informationstechnische Grundlagen

3.3.4 Informationsgesellschaft und Datensicherheit

Schülerinnen und Schüler nutzen im Alltag Technologien (zum Beispiel Identifikations- und Bezahlsysteme, Multifaktorauthentifizierung, Signaturen), denen kryptologische Systeme zugrunde liegen. Sie lernen den prinzipiellen Unterschied zwischen symmetrischen und asymmetrischen Verfahren kennen und erläutern, wie Infrastrukturen zur verschlüsselten Kommunikation und Authentifizierung mittels digitaler Zertifikate genutzt werden.

Informatiksysteme wie Datenbanksysteme und Netzwerkinfrastrukturen sind potenzielle Angriffsziele für Unbefugte, um Daten Dritter zu erlangen.

Die Schülerinnen und Schüler können gängige Angriffe (zum Beispiel SQL-Injection, DDOS, Phishing) beschreiben, um sowohl Sicherheitsrisiken als auch geeignete Maßnahmen zur Gewährleistung von Datensicherheit und Datenschutz zu diskutieren.

Die Schülerinnen und Schüler können

(1) <i>asymmetrische</i> und <i>symmetrische Verschlüsselung</i> vergleichen (Schlüsselverwaltung, Schlüsseltausch, Geschwindigkeit)
<p>P 2.2 Modellieren und Implementieren 1</p> <p>P 2.4 Analysieren und Bewerten 3</p>
(2) erklären, wie Nachrichten mit <i>asymmetrischer Verschlüsselung signiert</i> werden können
P 2.3 Kommunizieren und Kooperieren 2
(3) erläutern, dass die Sicherheit <i>asymmetrischer Kryptoverfahren</i> auf <i>Einwegfunktionen</i> basiert, für die ausschließlich Angriffe mit exponentieller Laufzeit bekannt sind
(4) spezielle Anforderungen an <i>kryptologische Hashfunktionen</i> und Anwendungen in der Kryptologie erläutern (zum Beispiel Fingerprint, Signatur, Passworhashes)
<p>P 2.3 Kommunizieren und Kooperieren 2</p> <p>I 3.3.1.1 Bitebene (9)</p>
(5) Kryptographische Ziele (Geheimhaltung, Authentifizierung, Integritätssicherung) und ihre Umsetzung mit kryptographischen Methoden (<i>Chiffren, Signaturen, Hashfunktionen</i>) erläutern
<p>P 2.3 Kommunizieren und Kooperieren 2</p> <p>I 3.3.1.1 Bitebene (9)</p>
(6) Anwendungsbereiche von <i>Public-Key-Infrastrukturen</i> beschreiben (zum Beispiel digitaler Personalausweis, Transportverschlüsselung, Mailverschlüsselung) und im Hinblick auf Sicherheitsaspekte bewerten
<p>P 2.3 Kommunizieren und Kooperieren 9, 10</p> <p>P 2.4 Analysieren und Bewerten 5, 8</p>
(7) erläutern, dass <i>digitale Zertifikate</i> auf Vertrauensmodellen (zum Beispiel hierarchisches Modell, Web of Trust) basieren
(8) Angriffe auf Informatiksysteme beschreiben (zum Beispiel SQL-Injections auf Datenbanksysteme, DDOS auf Netzwerkinfrastrukturen oder Phishing auf Zugangskontrollsysteme)
<p>P 2.3 Kommunizieren und Kooperieren 2, 8, 9</p> <p>P 2.4 Analysieren und Bewerten 5, 8</p> <p>I 3.3.1.3 Relationale Datenbanksysteme (8), (9)</p>
(9) Maßnahmen zur Gewährleistung von <i>Datensicherheit</i> und <i>Datenschutz</i> beim Einsatz von Informatiksystemen erläutern
<p>P 2.4 Analysieren und Bewerten 1, 3, 4, 7, 8, 9</p> <p>L BTV Minderheitenschutz; Wertorientiertes Handeln</p>
(10) Szenarien bewerten, in denen Daten massenweise erhoben, gespeichert und weiterverarbeitet werden
<p>P 2.4 Analysieren und Bewerten 5, 8, 9</p> <p>I 3.3.1.3 Relationale Datenbanksysteme</p> <p>L BTV Minderheitenschutz; Wertorientiertes Handeln</p>

- BNE Teilhabe, Mitwirkung, Mitbestimmung; Werte und Normen in Entscheidungssituationen
- BTV Minderheitenschutz; Toleranz, Solidarität, Inklusion, Antidiskriminierung
- MB Information und Wissen; Informationelle Selbstbestimmung und Datenschutz; Informationstechnische Grundlagen; Kommunikation und Kooperation; Mediengesellschaft
- PG Sicherheit und Unfallschutz
- VB Verbraucherrechte

3.3.5 Automaten und formale Sprachen

Im Alltag begegnen den Schülerinnen und Schülern Phänomene, die sich mithilfe formaler Sprachen beschreiben lassen (zum Beispiel KFZ-Kennzeichen, Rechenausdrücke, Zahlenschreibweisen, E-Mail-Adresse, URL, Geokoordinaten, XML, Programmiersprachen, CNC-Fräscodes, Chemische Verbindungen, Musiknotationen). Die Schülerinnen und Schüler entwickeln ausgehend von diesen Beispielen eine Vorstellung von formalen Sprachen und Kompetenzen im Umgang mit diesen (zum Beispiel Ableiten von Wörtern, Wortproblem, Grammatiken).

Die Schülerinnen und Schüler können Unterschiede und Grenzen von regulären und kontextfreien Sprachen erläutern und diese in der Chomsky-Hierarchie verorten. Sie nutzen unterschiedliche Darstellungsformen (zum Beispiel Automaten, Syntaxdiagramme, Produktionsregeln), um die jeweiligen Grammatiken zu beschreiben. Bei der Modellierung realer Automaten (zum Beispiel Leergutautomaten, Snackautomaten und Ticketautomaten) werden im Unterricht auch Automaten mit Ausgaben behandelt.

Die Schülerinnen und Schüler können

Formale Sprachen	
(1)	Anwendungsbereiche beschreiben, in denen <i>formale Sprachen</i> und Automatenmodelle eingesetzt werden
	<ul style="list-style-type: none"> P 2.2 Modellieren und Implementieren 4, 5 P 2.3 Kommunizieren und Kooperieren 8 P 2.4 Analysieren und Bewerten 1 I 3.3.3 Rechner und Netze (12), (13) L MB Information und Wissen; Produktion und Präsentation
(2)	die Definition einer <i>formalen Sprache</i> erläutern
	L MB Kommunikation und Kooperation
(3)	den Unterschied zwischen <i>Semantik</i> und <i>Syntax</i> einer <i>formalen Sprache</i> erläutern
	L MB Information und Wissen; Kommunikation und Kooperation
(4)	<i>Syntaxdiagramme</i> unter Verwendung von <i>Terminalsymbolen</i> und <i>Nichtterminalsymbolen</i> zur Beschreibung einer <i>formalen Sprache</i> entwerfen
	<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1
(5)	eine <i>Grammatik</i> unter Verwendung von <i>Terminalsymbolen</i> , <i>Nichtterminalsymbolen</i> , <i>Startsymbol</i> und <i>Produktionsregeln</i> (auch unter Verwendung von <i>EBNF</i>) zur Beschreibung einer <i>formalen Sprache</i> angeben
	<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 8 P 2.3 Kommunizieren und Kooperieren 1
(6)	für verschiedene Sprachtypen überprüfen, ob ein Wort einer <i>formalen Sprache</i> angehört (<i>Wortproblem</i>), eine <i>Ableitung</i> für ein Wort der Sprache angeben und Wörter der Sprache angeben
Reguläre und kontextfreie Sprachen und ihre Automaten	
(7)	einen <i>endlichen Automaten</i> mit und ohne Ausgabe (unter anderem <i>Mealy-Automat</i> als <i>Zustandsdiagramm</i>) zur Modellierung von Abläufen verwenden
	<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1
(8)	zu einem <i>endlichen Automaten</i> die Darstellungsformen <i>Zustandsdiagramm</i> und <i>Zustandsübergangstabelle</i> ineinander überführen
	<ul style="list-style-type: none"> P 2.1 Strukturieren und Vernetzen 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1
(9)	die Definition einer <i>regulären Sprache</i> erläutern

<p>P 2.3 Kommunizieren und Kooperieren 1</p>
<p>(10) <i>reguläre Sprachen</i> durch <i>deterministische endliche Automaten (DEA)</i>, <i>reguläre Grammatiken</i> und <i>reguläre Ausdrücke</i> beschreiben</p>
<p>P 2.1 Strukturieren und Vernetzen 8 P 2.3 Kommunizieren und Kooperieren 1</p>
<p>(11) die Grenzen <i>regulärer Sprachen</i> erläutern</p>
<p>P 2.4 Analysieren und Bewerten 6</p>
<p>(12) die Definition einer <i>kontextfreien Sprache</i> angeben</p>
<p>P 2.1 Strukturieren und Vernetzen 8 P 2.3 Kommunizieren und Kooperieren 1</p>
<p>(13) <i>kontextfreie Sprachen</i> durch <i>Kellerautomaten</i> und <i>kontextfreie Grammatiken</i> beschreiben</p>
<p>P 2.1 Strukturieren und Vernetzen 8 P 2.3 Kommunizieren und Kooperieren 1</p>
<p>(14) Grenzen <i>kontextfreier Sprachen</i> erläutern</p>
<p>P 2.4 Analysieren und Bewerten 6</p>
<p>(15) Einsatzbereiche für <i>reguläre Automaten</i> (zum Beispiel Substringproblem, GUI, zustandsbasierte Protokolle) und <i>Kellerautomaten</i> (Parsen von Klammersprachen) erläutern</p>
<p>P 2.3 Kommunizieren und Kooperieren 8 I 3.3.3 Rechner und Netze (13)</p>
<p>F D 3.3.2.1 Struktur von Äußerungen F D 3.4.2.1 Struktur von Äußerungen L MB Informationstechnische Grundlagen</p>

3.3.6 Projektmanagement

Komplexere Projekte lassen sich heute nur noch in Teams planen und realisieren. Zur Planung und Durchführung haben sich verschiedene Methoden etabliert. Die Schülerinnen und Schüler lernen Vorgehensmodelle und gängige Begriffe des Projektmanagements kennen.

Bei der Realisierung eigener Projekte mit dem Ziel, funktionsfähige Prototypen zu erstellen, wenden sie Methoden des Projektmanagements von der Planung über die Durchführung bis zur Dokumentation an. Dabei vertiefen Sie die erworbenen inhaltsbezogenen und prozessbezogenen Kompetenzen und vernetzen Kompetenzen aus mehreren Inhaltsbereichen miteinander.

So könnte ein Softwareprojekt eine Datenbank-Schnittstelle nutzen, um Anwendungs- und Nutzerdaten zu speichern, eine formale Sprache zur Speicherung eines Spielstands verwenden, mithilfe eines Automatenmodells die Logik einer GUI modellieren oder kryptographische Verfahren zur sicheren Übertragung einsetzen.

Die Schülerinnen und Schüler können

(1) Aspekte des Projektmanagements (unter anderem <i>Projekt, Projektauftrag, Projektphasen, Anforderungen, Abnahme</i>) und ein Vorgehensmodell (zum Beispiel agiles Vorgehen, iterativ, Wasserfallmodell) erklären
<p>P 2.1 Strukturieren und Vernetzen 5, 6, 7, 8, 9</p> <p>P 2.3 Kommunizieren und Kooperieren 5</p> <p>P 2.4 Analysieren und Bewerten 7</p>
(2) Vorgehensweisen und Herausforderungen bei der kollaborativen Softwareentwicklung (zum Beispiel Nutzung von Bibliotheken und APIs, Versionsverwaltung, Bereitstellung selbst erstellter Bibliotheken, Bugtracking, Updates) beschreiben
<p>P 2.1 Strukturieren und Vernetzen 1, 6, 7, 9</p> <p>P 2.3 Kommunizieren und Kooperieren 3, 4, 5, 6, 7, 8, 9</p> <p>I 3.3.2.1 Strukturierte und objektorientierte Programmierung (10), (21), (22)</p>
(3) Methoden testgetriebener oder testgestützter Entwicklung beschreiben und verwenden
<p>P 2.1 Strukturieren und Vernetzen 7</p> <p>P 2.2 Modellieren und Implementieren 2, 11, 12, 13</p>
(4) einen Projektentwurf erstellen und Anforderungen an das <i>Projekt</i> in geeigneter Form (zum Beispiel Lastenheft, Pflichtenheft, User Stories, Systemfunktionen) beschreiben, ein (Teil-)Projekt durchführen und einen Prototyp erstellen
<p>P 2.1 Strukturieren und Vernetzen 5, 6, 7, 8, 9</p> <p>P 2.2 Modellieren und Implementieren 2, 3, 4, 5, 8, 9</p> <p>P 2.3 Kommunizieren und Kooperieren 2, 5, 6, 7, 9, 10</p> <p>P 2.4 Analysieren und Bewerten 3, 4, 7</p> <p>L VB Bedürfnisse und Wünsche</p>
<p>L BNE Demokratiefähigkeit; Teilhabe, Mitwirkung, Mitbestimmung</p> <p>L BO Einschätzung und Überprüfung eigener Fähigkeiten und Potenziale; Fachspezifische und handlungsorientierte Zugänge zur Arbeits- und Berufswelt; Geschlechtsspezifische Aspekte bei der Berufswahl, Familien- und Lebensplanung; Kompetenzanalyse, Eignungstests und Entscheidungstrainings</p> <p>L BTV Konfliktbewältigung und Interessenausgleich; Selbstfindung und Akzeptanz anderer Lebensformen; Wertorientiertes Handeln</p> <p>L MB Information und Wissen; Kommunikation und Kooperation; Produktion und Präsentation</p> <p>L PG Selbstregulation und Lernen; Sicherheit und Unfallschutz</p> <p>L VB Bedürfnisse und Wünsche; Qualität der Konsumgüter; Umgang mit eigenen Ressourcen</p>

4. Operatoren

In den Standards für inhaltsbezogene Kompetenzen werden Operatoren (handlungsleitende Verben) verwendet. Standards legen fest, welche Anforderungen die Schülerinnen und Schüler in der Regel erfüllen. Zusammen mit der Zuordnung zu einem der drei Anforderungsbereiche (AFB) dienen Operatoren einer Präzisierung. Dies sichert das Erreichen des vorgesehenen Niveaus und die angemessene Interpretation der Standards.

Beschreibung der drei Anforderungsbereiche

- **Anforderungsbereich I** umfasst das Wiedergeben von Sachverhalten und Kenntnissen sowie das Anwenden und Beschreiben geübter Arbeitstechniken und Verfahren.
- **Anforderungsbereich II** umfasst das selbstständige Verarbeiten und Darstellen bekannter Sachverhalte in einem durch Übung bekannten Zusammenhang und das selbstständige Übertragen des Gelernten auf vergleichbare, neue Sachverhalte.
- **Anforderungsbereich III** umfasst das Verarbeiten komplexer Sachverhalte mit selbstständiger Auswahl geeigneter Arbeitstechniken mit dem Ziel, zu selbstständigen Lösungen, Gestaltungen oder Deutungen, Folgerungen, Verallgemeinerungen, Begründungen und Wertungen zu gelangen und das eigene Vorgehen zu reflektieren.

Zuordnung zu Anforderungsbereichen

Die Zuordnung eines Operators ist im Einzelfall auch vom Kontext der Aufgabenstellungen und ihrer unterrichtlichen Einordnung abhängig. Im Folgenden werden die Operatoren dem überwiegend in Betracht kommenden Anforderungsbereich zugeordnet.

Operatoren	Beschreibung	AFB
analysieren	eine konkrete Materialgrundlage unter einer gegebenen Fragestellung auf wichtige Bestandteile, Eigenschaften oder Zusammenhänge untersuchen	III
angeben	Ergebnisse numerisch oder verbal formulieren, ohne Darstellung des Lösungsweges und ohne Begründungen	I
anwenden, nutzen, umgehen mit, verwenden	Fachbegriffe, Regeln, mathematische Sätze, Zusammenhänge oder Verfahren auf einen (anderen) Sachverhalt beziehen	II
begründen	eine Aussage oder einen Sachverhalt durch Berechnungen, nach gültigen Schlussregeln, durch Herleitungen oder inhaltliche Argumentation verifizieren oder falsifizieren	III
berechnen	Ergebnisse von einem Ansatz oder einer Formel ausgehend durch Rechenoperationen gewinnen	I
beschreiben	Strukturen, Sachverhalte, Verfahren, Prozesse und Eigenschaften von Objekten in der Regel unter Verwendung der Fachsprache in vollständigen Sätzen wiedergeben (hier sind auch Einschränkungen möglich: „Beschreiben Sie in Stichworten“) beziehungsweise in einer vorgeschriebenen Form darstellen (zum Beispiel: „Beschreiben Sie als Term“)	II
bestimmen, ermitteln, erschließen	Lösungen, Lösungswege beziehungsweise Zusammenhänge auf der Basis von Vorkenntnissen oder Verfahren rechnerisch, grafisch oder experimentell finden und darstellen	II
bewerten	einen Sachverhalt nach fachwissenschaftlichen oder fachmethodischen Kriterien, persönlichem oder gesellschaftlichem Wertebezug begründet einschätzen und ein selbstständiges Urteil formulieren	III

darstellen	Zusammenhänge, Sachverhalte oder Arbeitsverfahren in strukturierter oder formal definierter Form (zum Beispiel grafisch) wiedergeben	II
durchführen	nach bekannten Regeln oder Anweisungen von einer Aufgabenstellung zu einem definierten Ziel gelangen	II
entwerfen	nach vorgegebenen Bedingungen ein sinnvolles Konzept selbstständig planen/erarbeiten	III
erklären	Sachverhalte, Strukturen, Prozesse und Zusammenhänge erfassen sowie auf Vorkenntnisse oder allgemeine Aussagen und Gesetze unter Verwendung der Fachsprache zurückführen	II
erläutern	Sachverhalte, Strukturen, Prozesse und Zusammenhänge erfassen sowie auf Vorkenntnisse oder allgemeine Aussagen und Gesetze unter Verwendung der Fachsprache zurückführen und durch zusätzliche Informationen oder Beispiele verständlich machen	II
erstellen	Herstellen und Gestalten eines Systems unter vorgegebener Zielsetzung	II
ergänzen, erweitern	weitere Bestandteile zu einem gegebenen Sachverhalt hinzufügen	II
identifizieren	Objekte, Muster oder Strukturen und die zugehörigen Fachbegriffe begründet miteinander verbinden	I
implementieren	Datenstrukturen oder Algorithmen in einer Programmiersprache umsetzen	II
interpretieren	Sachverhalte und Zusammenhänge im Hinblick auf Erklärungsmöglichkeiten untersuchen und abwägend herausstellen	III
kommentieren	einen gegebenen Sachverhalt oder einen gegebenen Algorithmus mit erläuternden Hinweisen versehen	I
modellieren	zu einem Ausschnitt der Realität ein informatisches Modell erstellen	II
nennen	Elemente, Sachverhalte, Begriffe, Daten, Fakten ohne Erläuterung wiedergeben	I
überführen	eine Darstellungsform in eine andere Darstellungsform bringen	II
überprüfen	durch Anwendung fachlicher Regeln oder Kenntnisse in einer ergebnisoffenen Situation einen vorgegebenen Sachverhalt verifizieren oder falsifizieren	III
untersuchen	Objekte, Sachverhalte und Fragestellungen nach fachlichen Kriterien zielorientiert erkunden und Zusammenhänge herausarbeiten	II
vergleichen	Gemeinsamkeiten, Ähnlichkeiten und Unterschiede herausarbeiten	II
zuordnen	einen begründeten Zusammenhang zwischen Objekten, Strukturen und Darstellungen herstellen	II

5. Anhang

5.1 Verweise

Das Verweissystem im Bildungsplan 2016 unterscheidet zwischen vier verschiedenen Verweisarten. Diese werden durch unterschiedliche Symbole gekennzeichnet:



Die vier verschiedenen Verweisarten

Die Darstellungen der Verweise weichen im Web und in der Druckfassung voneinander ab.

Darstellung der Verweise auf der Online-Plattform

Verweise auf Teilkompetenzen werden unterhalb der jeweiligen Teilkompetenz als anklickbare Symbole dargestellt. Nach einem Mausklick auf das jeweilige Symbol werden die Verweise im Browser detaillierter dargestellt (dies wird in der Abbildung nicht veranschaulicht):



Darstellung der Verweise in der Webansicht (Beispiel aus Geographie 3.1.2.1 „Grundlagen von Wetter und Klima“)

Darstellung der Verweise in der Druckfassung

In der Druckfassung und in der PDF-Ansicht werden sämtliche Verweise direkt unterhalb der jeweiligen Teilkompetenz dargestellt. Bei Verweisen auf andere Fächer ist zusätzlich das Fächerkürzel dargestellt (im Beispiel „BNT“ für „Biologie, Naturphänomene und Technik (BNT)“):



Darstellung der Verweise in der Druckansicht (Beispiel aus Geographie 3.1.2.1 „Grundlagen von Wetter und Klima“)

Gültigkeitsbereich der Verweise

Sind Verweise nur durch eine gestrichelte Linie von den darüber stehenden Kompetenzbeschreibungen getrennt, beziehen sie sich unmittelbar auf diese.

Stehen Verweise in der letzten Zeile eines Kompetenzbereichs und sind durch eine durchgezogene Linie von diesem getrennt, so beziehen sie sich auf den gesamten Kompetenzbereich.



Gültigkeitsbereich von Verweisen (Beispiel aus Ethik 3.1.2.2 „Verantwortung im Umgang mit Konflikten und Gewalt“)

5.2 Abkürzungen

Leitperspektiven

Allgemeine Leitperspektiven	
BNE	Bildung für nachhaltige Entwicklung
BTV	Bildung für Toleranz und Akzeptanz von Vielfalt
PG	Prävention und Gesundheitsförderung
Themenspezifische Leitperspektiven	
BO	Berufliche Orientierung
MB	Medienbildung
VB	Verbraucherbildung

Fächer des Gymnasiums

Abkürzung	Fach
ASTRO	Astronomie – Wahlfach in der Oberstufe
BIO	Biologie
BIO.V2	Biologie – Überarbeitete Fassung vom 08.03.2022
BK	Bildende Kunst
BKPROFIL	Bildende Kunst – Profulfach
BMB	Basiskurs Medienbildung

BNT	Biologie, Naturphänomene und Technik (BNT)
CH	Chemie
CH.V2	Chemie – Überarbeitete Fassung vom 25.03.2022
CHIN4	Chinesisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
D	Deutsch
DG	Darstellende Geometrie – Wahlfach in der Oberstufe
DMW	Digitale mathematische Werkzeuge – Wahlfach in der Oberstufe
E1	Englisch als erste Fremdsprache
E2	Englisch als zweite Fremdsprache
ETH	Ethik
F1	Französisch als erste Fremdsprache
F2	Französisch als zweite Fremdsprache
F3	Französisch als dritte Fremdsprache – Profulfach
F4	Französisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
G	Geschichte
GEO	Geographie
GEOL	Geologie – Wahlfach in der Oberstufe
GK	Gemeinschaftskunde
GR3	Griechisch als dritte Fremdsprache – Profulfach
GR4	Griechisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
HEBR4	Hebräisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
IMP	Informatik, Mathematik, Physik (IMP) – Profulfach
INF	Informatik
INFWFO	Informatik – Wahlfach in der Oberstufe
INF7	Aufbaukurs Informatik (Klasse 7)
ITAL3	Italienisch als dritte Fremdsprache – Profulfach
ITAL4	Italienisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
JAP4	Japanisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
L1	Latein als erste Fremdsprache
L2	Latein als zweite Fremdsprache

L3	Latein als dritte Fremdsprache – Profulfach
L4	Latein als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
LIT	Literatur – Wahlfach in der Oberstufe
LUT	Literatur und Theater – Wahlfach in der Oberstufe
M	Mathematik
MUS	Musik
MUSPROFIL	Musik – Profulfach
NWT	Naturwissenschaft und Technik (NwT) – Profulfach
PH	Physik
PH.V2	Physik – Überarbeitete Fassung vom 25.03.2022
PHIL	Philosophie – Wahlfach in der Oberstufe
PORT3	Portugiesisch als dritte Fremdsprache – Profulfach
PORT4	Portugiesisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
PSY	Psychologie – Wahlfach in der Oberstufe
RAK	Altkatholische Religionslehre
RALE	Alevitische Religionslehre
REV	Evangelische Religionslehre
RISL	Islamische Religionslehre sunnitischer Prägung
RJUED	Jüdische Religionslehre
RORTH	Orthodoxe Religionslehre
RRK	Katholische Religionslehre
RSYR	Syrisch-Orthodoxe Religionslehre
RU2	Russisch als zweite Fremdsprache
RU3	Russisch als dritte Fremdsprache – Profulfach
RU4	Russisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
SPA3	Spanisch als dritte Fremdsprache – Profulfach
SPA4	Spanisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe
SPO	Sport
SPOPROFIL	Sport – Profulfach
TUERK4	Türkisch als spät beginnende Fremdsprache – Wahlfach in der Oberstufe

WBS	Wirtschaft / Berufs- und Studienorientierung (WBS)
WI	Wirtschaft

5.3 Geschlechtergerechte Sprache

Im Bildungsplan 2016 wird in der Regel durchgängig die weibliche Form neben der männlichen verwendet; wo immer möglich, werden Paarformulierungen wie „*Lehrerinnen und Lehrer*“ oder neutrale Formen wie „*Lehrkräfte*“, „*Studierende*“ gebraucht.

Ausnahmen von diesen Regeln finden sich bei

- Überschriften, Tabellen, Grafiken, wenn dies aus layouttechnischen Gründen (Platzmangel) erforderlich ist,
- Funktions- oder Rollenbezeichnungen beziehungsweise Begriffen mit Nähe zu formalen und juristischen Texten oder domänenspezifischen Fachbegriffen (zum Beispiel „*Marktteilnehmer*“, „*Erwerbstätiger*“, „*Auftraggeber*“, „*(Ver)Käufer*“, „*Konsument*“, „*Anbieter*“, „*Verbraucher*“, „*Arbeitnehmer*“, „*Arbeitgeber*“, „*Bürger*“, „*Bürgermeister*“),
- massiver Beeinträchtigung der Lesbarkeit.

Selbstverständlich sind auch in all diesen Fällen Personen jeglichen Geschlechts gemeint.

5.4 Besondere Schriftauszeichnungen

Klammern und Verbindlichkeit von Beispielen

Im Fachplan sind einige Begriffe in Klammern gesetzt.

Steht vor den Begriffen in Klammern „zum Beispiel“, so dienen die Begriffe lediglich einer genaueren Klärung und Einordnung.

Begriffe in Klammern ohne „zum Beispiel“ sind ein verbindlicher Teil der Kompetenzformulierung.

Steht in Klammern ein „unter anderem“, so sind die in der Klammer aufgeführten Aspekte verbindlich zu unterrichten und noch weitere Beispiele der eigenen Wahl darüber hinaus.

Kursivschreibung

Fachbegriffe, die *kursiv* geschrieben sind, sind im Unterricht verbindlich mit dem Ziel einzusetzen, dass die Schülerinnen und Schüler diese

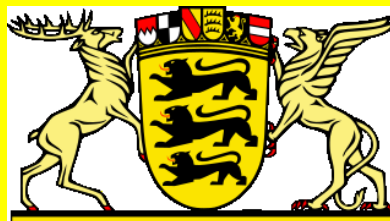
- in unterschiedlichen Kontexten ohne zusätzliche Erläuterung verstehen und anwenden können,
- im eigenen Wortschatz als Fachsprache aktiv benutzen können,
- mit eigenen Worten korrekt beschreiben können.

Fachbegriffe, die in den Standards *nicht kursiv* gesetzt sind, werden verwendet, um die Kompetenzbeschreibung für die Lehrkräfte fachlich präzise und prägnant formulieren zu können. Die Schülerinnen und Schüler müssen über diese Fachbegriffe nicht verfügen können.

Leerzeilen/Leerkompetenzen

Die Teilkompetenzen der inhaltsbezogenen Kompetenzen haben im Leistungsfach und im Basisfach die gleiche Nummerierung. Teilkompetenzen, die nur im Leistungsfach ausgewiesen sind, sind im Basisfach mit [nur LF] gekennzeichnet.

Ministerium für Kultus, Jugend und Sport
Postfach 103442, 70029 Stuttgart



www.bildungsplaene-bw.de