

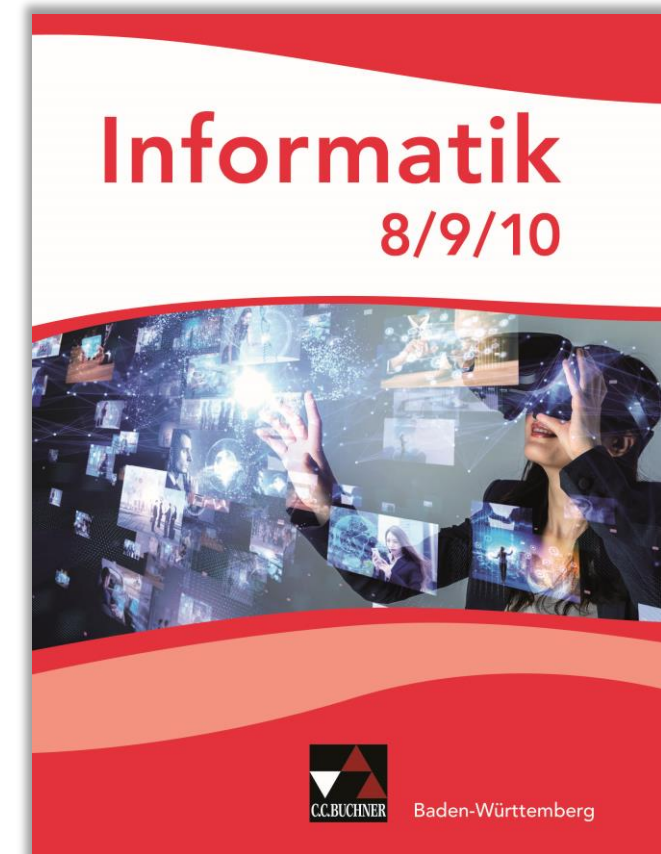
STOFFVERTEILUNGSPLAN ZUM GEMEINSAMEN BILDUNGSPLAN DER
SEKUNDARSTUFE I FÜR INFORMATIK, MATHEMATIK, PHYSIK (IMP) –
TEILBEREICH INFORMATIK

Informatik 8/9/10

Informatik – Baden-Württemberg

ISBN 978-3-661-38143-5

Stoffverteilungsplan auf
www.ccbuchner.de auch als
kostenfreier Download im
PDF- und Word-Format
erhältlich (Eingabe ins
Suchfeld: 38143).



Hinweis:

Die Kompetenzerwartungen und inhaltlichen Schwerpunkte wurden entsprechend des gemeinsamen Bildungsplans der Sekundarstufe I Baden-Württemberg vorgenommen. Die Inhaltsfelder und Kompetenzbereiche entsprechen den Ausführungen im Bildungsplan.

Inhaltbezogene Kompetenzen: Daten und Codierung; Algorithmen; Rechner und Netze; Informationsgesellschaft und Datensicherheit

Wenn nicht explizit ausgewiesen, sind die Niveaustufen der inhaltsbezogenen Kompetenzen identisch.

Übergeordnete Kompetenzbereiche:

<p>Strukturieren und Vernetzen (SV)</p> <p>Die Schülerinnen und Schüler können</p> <ol style="list-style-type: none"> 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.), 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). 	<p>Analysieren und Bewerten (AB)</p> <p>Die Schülerinnen und Schüler können</p> <ol style="list-style-type: none"> 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, 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.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Modellieren und Implementieren (MI)

Die Schülerinnen und Schüler können

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),
5. relevante Abläufe, Daten und ihre Beziehungen in informatischen Modellen darstellen,
6. passende Strukturen und Lösungsstrategien für gegebene Problemstellungen auswählen,
7. geeignete Programme und Hilfsmittel zur grafisch gestützten Modellierung einsetzen,
8. unterschiedliche Perspektiven in die Entwicklung einer Lösung miteinbeziehen,
9. Abläufe in einer (zum Beispiel grafischen) Programmiersprache implementieren,
10. geeignete Codebausteine aus verschiedenen Quellen auswählen, gegebenenfalls adaptieren und in eigene Programme einbauen,
11. Programme gezielt gegen vorab formulierte Testfälle testen,
12. Fehler in der Implementierung systematisch aufspüren und beheben (zum Beispiel Debugger),
13. die Angemessenheit von Lösungen und die erreichten Resultate bewerten.

Kommunizieren und Kooperieren (KK)

Die Schülerinnen und Schüler können

1. fachspezifische Schreib- und Notationsweisen verwenden,
2. Sachverhalte, eigene Ideen, Lösungswege und Ergebnisse zielgruppenorientiert und unter Beachtung der informatischen Terminologie erläutern und strukturiert darstellen,
3. eigenen und fremden Programmcode in geeigneter Weise kommentieren und dokumentieren,
4. vorhandene Dokumentationen und kommentierten Programmcode lesen und verstehen,
5. arbeitsteilig als Team ihre Aufgaben planen, strukturieren, ausführen, reflektieren und präsentieren,
6. zielorientiert auf einer vorhandenen Infrastruktur kommunizieren und geeignete digitale Werkzeuge zum Teilen von Informationen (zum Beispiel Arbeitsergebnisse, Fragen, Programmcode) einsetzen,
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.

1 Daten und Codierung Klasse 8 (ca. 4 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
1.1 Fehlererkennung S.8 1.2 Paritätsprüfung S.10	Die Schülerinnen und Schüler können <ul style="list-style-type: none"> ➤ die Begriffe Fehlerkorrektur und Fehlererkennung erklären und alltagsrelevanten Beispielen zuordnen (z.B. zerkratzte CD, unlesbarer Barcode), G ➤ die Begriffe Fehlerkorrektur, Fehlererkennung, Redundanz erklären und deren Notwendigkeit anhand von alltagsrelevanten Beispielen erläutern (z. B. zerkratzte CD, unlesbarer Barcode), ME ➤ Verfahren zur Fehlererkennung (u.a. Prüfsumme) anwenden (z.B. bei EAN, Personalausweisnummer), ➤ die Güte eines Fehlerkorrekturverfahrens hinsichtlich verschiedener Fehlertypen (z.B. Einbitfehler, Zweibitfehler, Zahlendreher) bewerten, M ➤ die Güte eines Fehlerkorrekturverfahrens hinsichtlich verschiedener Fehlertypen (z.B. Einbitfehler, Zweibitfehler, Zahlendreher), Speicherplatzbedarf und Einsatzbereich bewerten. E 	<ul style="list-style-type: none"> - AB3 - AB5

2 Algorithmen Klasse 8 (ca. 12 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
2.1 Logische Verknüpfungen in Scratch S.40	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> ➤ Logische Verknüpfungen (UND, ODER, NICHT) in Bedingungen von Schleifen und Verzweigungen verwenden, ➤ Zufallszahlen in eigenen Programmen verwenden (z.B. um Würfelergebnisse zu simulieren oder einen Spielverlauf abwechslungsreicher zu gestalten), ➤ eine indexbasierte Datenstruktur zur Speicherung und Verarbeitung gleichartiger Daten (auch per Iteration) verwenden, ➤ grundlegende Algorithmen auf einer indexbasierten Datenstruktur (z.B. Füllen mit Werten, Maximumsuche, Summenbildung) implementieren, G ➤ grundlegende Algorithmen auf einer indexbasierten Datenstruktur (z.B. Füllen mit Werten, Maximumsuche, Summenbildung) beschreiben und implementieren, M ➤ grundlegende Algorithmen auf einer indexbasierten Datenstruktur (z.B. Füllen mit Werten, Maximumsuche, Summenbildung, Bubblesort) erläutern und implementieren, E ➤ Anforderungen an Programme oder Programmteile beschreiben und die Implementierungen mithilfe vorgegebener Testfälle testen, 	<ul style="list-style-type: none"> - SV4 - SV6 - SV7 - MI4 - MI5 - MI9 - MI11 - MI12
2.2 Zufallszahlen in Scratch S.42		
2.3 Listen in Scratch S.44		
2.4 Datenstrukturen und Algorithmen S.46		
2.5 Unterprogramme in Scratch S.48		
2.6 Checkerseite: MakeCode Arcade S.50		
2.21 Mikrocontroller S.78		
2.22 Mikrocontroller blockbasiert programmieren S.80		
2.28 Projekte* S.92		

	<ul style="list-style-type: none">➤ ein kleines Softwareprojekt (z.B. Spiel, Smartphone-App, Robotik, Simulation) unter Anleitung durchführen.	
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------	--

*Neben einem allgemeinen Überblick zur Strukturierung von Projekten entlang des Entwicklungsprozesses beinhaltet das Kapitel einige konkrete Projektvorschläge. Schülerinnen und Schüler können einzelne dieser Projektvorschläge wie im Buch ausgearbeitet umsetzen, als Anstoß für ein abgewandeltes Projekt nehmen oder komplett frei eigene Projekte durchführen.

3 Rechner und Netze Klasse 8 (ca. 8 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
3.1 Lokales Rechnernetz S.110	Die Schülerinnen und Schüler können	- MI5
3.2 Einführung in die Netzwerksimulationsumgebung Filius S.112	<ul style="list-style-type: none"> ➤ den grundlegenden Aufbau eines lokalen Rechnernetzes und die Rolle seiner Komponenten (Endgerät, Verbindung, Verteiler) beschreiben, G 	- MI6
3.3 Paketorientierte Datenübertragung S.114	<ul style="list-style-type: none"> ➤ den grundlegenden Aufbau eines lokalen Rechnernetzes und die Rolle seiner Komponenten (Endgerät, Verbindung, Verteiler) erklären, ME 	- MI7
3.4 Adressierung S.116	<ul style="list-style-type: none"> ➤ die Notwendigkeit einer eindeutigen Adressierung zur Kommunikation in Netzen erläutern und hierfür Beispiele nennen (IP-Adresse und z.B. Handynummer, E-Mail-Adresse), 	- KK7
3.5 Protokolle S.118	<ul style="list-style-type: none"> ➤ das Prinzip der paketorientierten Datenübertragung erläutern, 	
3.6 Namensauflösung S.120	<ul style="list-style-type: none"> ➤ die Notwendigkeit eines Protokolls für technische und nicht-technische Kommunikation in einem gegebenen Szenario erläutern (z.B. Ping-Anfrage, moderiertes Gespräch in Gruppe), G 	
3.7 DNS in Filius S.122	<ul style="list-style-type: none"> ➤ die Notwendigkeit eines Protokolls für technische und nicht-technische Kommunikation erläutern (z.B. Ping-Anfrage, moderiertes Gespräch in Gruppe) und für geeignete Szenarien eigene Protokolle entwerfen, ME 	
3.8 Der eigene Webserver S.124	<ul style="list-style-type: none"> ➤ das Prinzip der Namensauflösung (DNS in einem lokalen Rechnernetz und z.B. Kontaktliste, Telefonbuch) erklären, M 	

	<ul style="list-style-type: none">➤ das Prinzip der Namensauflösung (DNS in einem lokalen Rechnernetz und z.B. Kontaktliste, Telefonbuch) erläutern, E➤ ein vorgegebenes lokales Rechnernetz mit Webserver in einer geeigneten Simulationsumgebung unter Anleitung erstellen, G➤ ein lokales Rechnernetz mit DNS und Webserver in einer geeigneten Simulationsumgebung unter Anleitung entwerfen und untersuchen, M➤ ein lokales Rechnernetz mit DNS und Webserver in einer geeigneten Simulationsumgebung entwerfen und untersuchen, E➤ einfache Webseiten mit Links (auch zu anderen Webservern) entwerfen. E	
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

4 Informationsgesellschaft und Datensicherheit Klasse 8 (ca. 4 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
4.1 Datensammler: Webtracking* S.144	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> ➤ eine aktuell eingesetzte Technologie erläutern, mit der personenbezogene Daten gesammelt werden (z.B. Webtracking, Cookies, Geodaten), ➤ Möglichkeiten beschreiben, um das Sammeln personenbezogener Daten einzuschränken (z.B. anonymes Surfen, Rechteverwaltung von Apps, Standortfreigabe), GM ➤ Möglichkeiten erläutern, um das Sammeln personenbezogener Daten einzuschränken (z.B. anonymes Surfen, Rechteverwaltung von Apps, Standortfreigabe). E 	<ul style="list-style-type: none"> - KK2 - KK7 - KK8 - KK9 - KK10
4.2 Datensammler: Cookies* S.146		
4.3 Datensammler einschränken: Anonymes Surfen** S.148		
4.4 Datensammler einschränken: App-Berechtigungen** S.150		
4.5 Datensammler einschränken: Standortfreigabe** S.152		

* Die Kapitel 4.1 und 4.2 behandeln jeweils aktuell eingesetzte Technologien zum Sammeln von personenbezogenen Daten und können alternativ verwendet werden. Es müssen nicht beide Kapitel behandelt werden.

** Die Kapitel 4.3, 4.4 und 4.5 behandeln jeweils Möglichkeiten das Sammeln personenbezogener Daten einzuschränken und können alternativ verwendet werden. Es müssen nicht alle drei Kapitel behandelt werden.

1 Daten und Codierung Klasse 9 (ca. 2 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
1.3 Listen, Bäume und Graphen S.12	Die Schülerinnen und Schüler können <ul style="list-style-type: none"> ➤ im Alltag die Strukturen Liste, Baum und Graph (z.B. Namensliste, Stammbaum, Organigramm, Straßenkarte, S-Bahn-Netz) identifizieren und mit diesen Strukturen Daten geeignet darstellen (z.B. Systematik im Tierreich, Schule). 	<ul style="list-style-type: none"> - AB3 - AB4

2 Algorithmen Klasse 9 (ca. 15 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
2.9 Entwicklungsumgebungen für Python S.54	Die Schülerinnen und Schüler können <ul style="list-style-type: none"> ➤ vorgegebene Programmbibliotheken für eigene Programme sinnvoll verwenden, ➤ Algorithmen mit den Grundbausteinen Anweisung, Bedingung, Schleife und Verzweigung sowie unter Verwendung von Variablen in einer geeigneten textuellen Programmiersprache implementieren, ➤ die Datentypen für Ganzzahl, Gleitkommazahl, Wahrheitswert und Zeichenkette beschreiben und anwenden, ➤ Programmcode sinnvoll kommentieren, ➤ syntaktische und semantische Fehler identifizieren, G ➤ den Unterschied zwischen syntaktischen und semantischen Fehlern erläutern, ME ➤ Programme auf semantische Fehler testen, ➤ Fehlermeldungen der Entwicklungsumgebung (z. B. Compilerfehler, Laufzeitfehler) nutzen, um Programme fehlerfrei zu implementieren, ➤ Strategien (z. B. Debugger, schrittweise Ausführung, Logging) anwenden, um das Verhalten von Programmcode zur Laufzeit zu beobachten, ➤ Unterprogramme – auch mit Parametern und Rückgabewerten – sinnvoll einsetzen, 	<ul style="list-style-type: none"> - AB1 - AB4 - SV2 - SV3 - SV5 - SV6 - SV7 - MI2 - MI4 - MI6 - MI8 - MI9 - MI10 - MI11 - MI12 - MI13 - KK1 - KK3 - KK4 - KK8
2.10 Bibliotheken S.56		
2.11 Schleifen S.58		
2.12 Datentypen S.60		
2.13 Verständlich programmieren S.62		
2.14 Syntax und Semantik S.64		
2.15 Fehlermeldung und Debugging S.66		
2.16 Unterprogramme S.68		
2.17 Zufallszahlen S.70		
2.18 Programme analysieren S.72		
2.19 Verzweigung S.74		
2.20 Unterprogramme planen S.76		
2.25 Checkerseite: Listen in Python S.84		

2.24 Checkerseite: Bubblesort S.104	<ul style="list-style-type: none">➤ Zufallszahlen in eigenen Programmen verwenden,➤ vorgegebenen Code auf dessen Funktionsweise hin analysieren und dessen Wirkung beschreiben,➤ Anpassungen zu vorgegebenem Code implementieren,➤ Anforderungen an Unterprogramme beschreiben, G➤ Anforderungen an Unterprogramme beschreiben und diese automatisiert (z. B. mit Testroutinen) testen, ME➤ Arrays zur Speicherung und Verarbeitung von Daten verwenden, E➤ grundlegende Algorithmen auf Arrays (z. B. Füllen mit Werten, Maximumsuche, Summenbildung, Bubblesort) erläutern und implementieren. E	
-----------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

4 Informationsgesellschaft und Datensicherheit Klasse 9 (ca. 10 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
4.15 Vigenère-Verschlüsselung S.168	Die Schülerinnen und Schüler können <ul style="list-style-type: none"> ➤ das Vigenère-Verfahren erklären und durchführen, ➤ das One-Time-Pad-Verfahren als absolut sicheres Verschlüsselungsverfahren erklären, M ➤ das One-Time-Pad-Verfahren erklären und begründen, dass es sich um ein absolut sicheres Verschlüsselungsverfahren handelt, E ➤ beschreiben, dass Verschlüsselungsverfahren ein Kompromiss zwischen Sicherheit und Praktikabilität sind, GM ➤ erläutern, dass moderne symmetrische Verschlüsselungsverfahren auf elementaren Verschlüsselungsverfahren basieren und ein Kompromiss zwischen Sicherheit und Praktikabilität sind, E ➤ erläutern, dass die Sicherheit von Verschlüsselungsverfahren nicht von der Geheimhaltung des Algorithmus abhängen darf (Kerckhoffs'sches Prinzip), ➤ Anwendungsbereiche beschreiben, in denen Verschlüsselung eingesetzt wird (z.B. verschlüsselte Speicherung von Daten, Kommunikation über https oder Messenger), ➤ die Verschlüsselung eigener Daten mithilfe eines geeigneten Programms durchführen. 	<ul style="list-style-type: none"> - AB5 - AB6 - AB7 - SV1 - KK8
4.16 One-Time-Pad-Verschlüsselung S.170		
4.17 Kerckhoffs' Prinzip S.174		
4.18 Anwendung von Verschlüsselung S.174		
4.19 Eigene Daten verschlüsseln S.176		

	<ul style="list-style-type: none">➤ Transpositionsverfahren (z. B. Skytale) und Substitutionsverfahren vergleichen, * GM➤ Transpositionsverfahren (z. B. Skytale), monoalphabetische Substitution und polyalphabetische Substitution vergleichen, * E➤ eine grundlegende Angriffsstrategie auf das Vigenère-Verfahren an einfachen Beispielen durchführen, * M➤ eine grundlegende Angriffsstrategie auf das Vigenère-Verfahren erklären und an einfachen Beispielen durchführen. * E	
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

* Zusatzmaterial zur Behandlung der inhaltsbezogenen Kompetenz im digitalen Lehrmaterial click & teach verfügbar

1 Daten und Codierung Klasse 10 (ca. 4 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
1.13 Lauflängencodierung S.28 1.14 Verlustbehaftete Datenkompression S.30 1.15 Verlustfreie Datenkompression S.32	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> ➤ erläutern, welche Parameter bei der Digitalisierung analoger Signale in Daten mittels Diskretisierung eine Rolle spielen (z.B. Wertebereich, Samplingtiefe, Abtastrate, Datenmenge, Datenrate), ME ➤ Verfahren zur Datenreduktion (z.B. Verringerung von Farbtiefe, Auflösung, Samplingtiefe, Samplingrate) beschreiben, ➤ die Lauflängencodierung als Beispiel für ein verlustfreies Datenkompressionsverfahren erläutern und an einem Beispiel händisch durchführen (z.B. Kompression von s/w-Bildern), ➤ Einsatzbereiche und Beispiele für verlustbehaftete und verlustfreie Datenkompressionsverfahren nennen (z.B. Kompression von Videos, Grafiken, Musik, Programmcode oder Textdokumenten). 	<ul style="list-style-type: none"> - AB2 - AB3 - KK5

2 Algorithmen Klasse 10 (ca. 6 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
1.8 Kurze Wege S.20	Die Schülerinnen und Schüler können <ul style="list-style-type: none"> ➤ Algorithmen entwerfen und implementieren, die zur Laufzeit Daten (z.B. Benutzereingaben oder Sensordaten) anfordern und auswerten, ➤ in eigenen Programmen Daten aus Dateien einlesen, verarbeiten und in Dateien schreiben (ggf. mittels geeigneter Bibliotheken) *, E ➤ erläutern, dass die Brute-Force-Methode für das Problem des kürzesten/ schnellsten Pfades in der Regel ungeeignet ist, GM ➤ das Problem des kürzesten/ schnellsten Pfades erläutern und begründen, dass die Brute-Force-Methode zur Lösung in der Regel ungeeignet ist, E ➤ den Algorithmus von Dijkstra zur Wegsuche an einem Beispiel durchführen, G ➤ den Algorithmus von Dijkstra zur Wegsuche erklären und händisch an einem Beispiel durchführen. ME 	<ul style="list-style-type: none"> - AB - SV5 - SV7 - MI9 - KK4
1.9 Der Algorithmus von Dijkstra S.22		
2.21 Mikrocontroller S.78		
2.22 Mikrocontroller blockbasiert programmieren S.80		
2.24 Mikrocontroller textbasiert programmieren S.82		

* Zusatzmaterial zur Behandlung der inhaltsbezogenen Kompetenz im digitalen Lehrmaterial click & teach verfügbar

3 Rechner und Netze Klasse 10 (ca. 5 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
3.12 Adressierung in Netzwerken S.110	Die Schülerinnen und Schüler können <ul style="list-style-type: none"> ➤ Schemata beschreiben, mit denen eine Unterscheidung von Adressen in Netzwerken nach lokal/global möglich ist (z.B. Subnetzmaske in IP-Netzen, Vorwahl im Telefonnetz, Länderkennung bei Postanschrift), ➤ einen Routingvorgang von Datenpaketen mithilfe eines geeigneten Werkzeugs (z.B. visuelles Traceroute) darstellen, G ➤ einen Routingvorgang von Datenpaketen mithilfe eines geeigneten Werkzeugs (z.B. visuelles Traceroute) darstellen und das Problem des Routings beschreiben, M ➤ das Problem des Routings zwischen Netzen erläutern und in einer geeigneten Simulationsumgebung ein Routingszenario durchführen, E ➤ das Prinzip der Namensauflösung von globalen Domainnamen erklären und in einer geeigneten Simulationsumgebung ein Namensauflösungsszenario durchführen. ME 	<ul style="list-style-type: none"> - AB - SV - MI - KK
3.13 Subnetting S.112		
3.14 Routing S.114		
3.15 DNS S.116		

4 Informationsgesellschaft und Datensicherheit Klasse 10 (ca. 5 Stunden)

Informatik 8/9/10	Inhaltsbezogene Kompetenzen	Übergeordnete Kompetenzerwartungen
4.9 Gründe für Datenverlust S.158	<p>Die Schülerinnen und Schüler können</p> <ul style="list-style-type: none"> ➤ Gründe nennen, die zu Datenverlust führen können (z.B. Verlust des Datenträgers, physikalischer Defekt, Schadsoftware, Fehlfunktion von Software, versehentliches Löschen), ➤ verschiedene Arten der Datensicherung (Vollbackup, inkrementelles Backup, differentielles Backup) beschreiben, ➤ eigene Backupstrategien entwickeln. 	<ul style="list-style-type: none"> - AB3 - AB5
4.10 Arten von Datensicherung S.160		
4.11 Eigene Backup-Strategien S.162		

Sprachsensibler Fachunterricht

Formen von sprachsensiblen Fachunterricht	Beispiele zur Umsetzung in Informatik 8/9/10
Erklärung von Fachbegriffen	Die wichtigsten Fachbegriffe des Buches mit passender Erklärung finden sich im Glossar ab Seite 186
Wortspeicher am Abschnittsende	Die wichtigsten Fachbegriffe eines Abschnittes finden sich auf den Seiten Alles im Blick: S.19 S.27 S.38 S.53 S.88 S.108 S.129 S.142 S.157 S.167 S.181
Hinweise in der Randspalte	An einigen Stellen des Buches finden sich u.a. auch Sprachhilfen in der Randspalte.