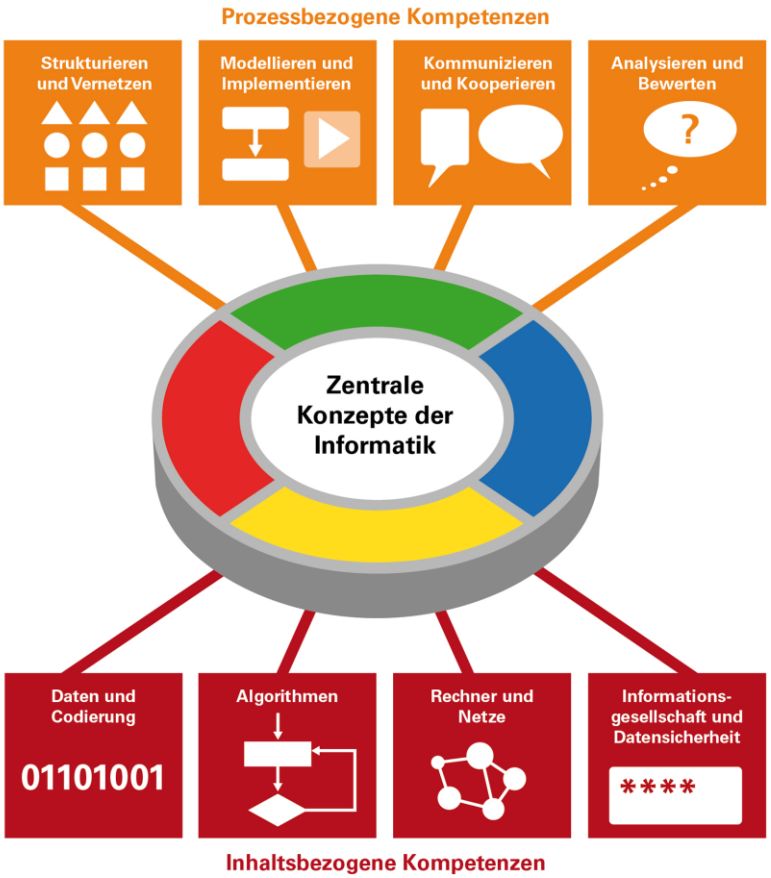

Prozessbezogene Kompetenzen

Die prozessbezogenen Kompetenzen gliedern sich in vier Kompetenzbereiche:

- Strukturieren und Vernetzen
- Modellieren und Implementieren
- Kommunizieren und Kooperieren
- Analysieren und Bewerten

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

Inhaltsbezogene Kompetenzen

3.1.1 Daten und Codierung

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

Daten und Codierung
(1) die Überführung von Zahlen zwischen ihrer Darstellung im *Hexadezimalzahlsystem*, *Binärsystem* und *Dezimalsystem* durchführen
F IMP 3.1.2.1 Mathematische Grundlagen der Kryptologie (2), (3) F INF 3.1.1 Daten und Codierung (1) L MB Informationstechnische Grundlagen
(2) Addition und Subtraktion von Binärzahlen in *Zweierkomplementdarstellung* schriftlich durchführen und das Ergebnis interpretieren
P 2.3 Kommunizieren und Kooperieren 1 P 2.4 Analysieren und Bewerten 3 F INF 3.2.1.1 Bitebene (2)
(3) die Begriffe *Fehlerkorrektur*, *Fehlererkennung*, *Redundanz* erklären und deren Notwendigkeit erläutern
P 2.3 Kommunizieren und Kooperieren 1 P 2.4 Analysieren und Bewerten 3, 5 F IMP 3.1.1.1 Daten und Codierung (1) L PG Selbstregulation und Lernen
(4) Verfahren zur *Fehlererkennung* (unter anderem *Prüfsumme*) anwenden
F IMP 3.1.1.1 Daten und Codierung (2)
(5) Verfahren zur *Datenreduktion* beziehungsweise *verlustbehafteten Datenkompression* (zum Beispiel Verringerung von Farbtiefe, Auflösung, Samplingtiefe, Samplingrate) beschreiben
F IMP 3.2.1.1 Daten und Codierung (2) F INF 3.1.1 Daten und Codierung (2) L MB Information und Wissen
(6) die *Lauflängencodierung* als Beispiel für ein *verlustfreies Datenkompressionsverfahren* erläutern und an einem Beispiel händisch durchführen (zum Beispiel Kompression von s/w-Bildern)
F IMP 3.2.1.1 Daten und Codierung (3) F INF 3.1.1 Daten und Codierung (3)
(7) Einsatzbereiche und Beispiele für *verlustbehaftete* und *verlustfreie Datenkompressionsverfahren* nennen (zum Beispiel Kompression von Videos, Grafiken, Musik, Programmcode oder Textdokumenten)
F IMP 3.2.1.1 Daten und Codierung (4) F INF 3.1.1 Daten und Codierung (4) L PG Selbstregulation und Lernen
L MB Information und Wissen; Informationstechnische Grundlagen
Relationale Datenbanksysteme
(8) Komponenten (unter anderem *Datenbank*, *Datenbankschnittstelle*, *Datenbankverwaltungssystem*) und deren Funktionen von *Datenbanksystemen* beschreiben
F INF 3.2.1.3 Relationale Datenbanksysteme (1)
(9) das relationale Datenbankmodell und wesentliche Begriffe (*Entität*, *Entitätstyp*, *Attribut*, *Relation*, *Kardinalität*, *Tabelle*, *Datensatz*, *Datenfeld*) beschreiben
F INF 3.2.1.3 Relationale Datenbanksysteme (2)
(10) die Begriffe *Schlüssel*, *Primärschlüssel* und *Fremdschlüssel* erläutern
F INF 3.2.1.3 Relationale Datenbanksysteme (3)
(11) Ausschnitte der Realität in einem Diagramm (Entity-Relationship-Diagramm oder UML-Klassendiagramm) modellieren
P 2.1 Strukturieren und Vernetzen 2, 8 P 2.2 Modellieren und Implementieren 5, 7, 13 P 2.3 Kommunizieren und Kooperieren 1 I 3.1.2 Algorithmen (11) F INF 3.2.1.3 Relationale Datenbanksysteme (4)
(12) ein *relationales Datenbankschema* in 3. Normalform erstellen, um unerwünschte *Redundanzen* zu vermeiden und damit das Risiko für Inkonsistenzen zu minimieren
P 2.1 Strukturieren und Vernetzen 8 P 2.2 Modellieren und Implementieren 5, 7 P 2.3 Kommunizieren und Kooperieren 1 F INF 3.2.1.3 Relationale Datenbanksysteme (5) L MB Produktion und Präsentation
(13) ein *Datenbankverwaltungsprogramm* zur Erstellung und Administration von *Datenbanken* nutzen
F INF 3.2.1.3 Relationale Datenbanksysteme (7)
(14) Abfragen auf *Datenbanken* in der *Datenbanksprache SQL* (Projektion, Selektion und Verbund über WHERE) auch über mehrere Tabellen durchführen
P 2.2 Modellieren und Implementieren 3, 9, 13 P 2.3 Kommunizieren und Kooperieren 1 P 2.4 Analysieren und Bewerten 4 F INF 3.2.1.3 Relationale Datenbanksysteme (8)
L MB Information und Wissen; Informationstechnische Grundlagen

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. Für spezielle Aufgabenbereiche (zum Beispiel Ein- und Ausgabe von Text, Grafik, Sensorabfragen) können geeignete Programmbibliotheken zum Einsatz kommen.
Mit der Objektorientierten Programmierung (OOP) lernen die Schülerinnen und Schüler ein weiteres 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.
Sie implementieren interaktive Programme mit einer einfachen grafischen Benutzerschnittstelle.
Die SchülerInnen und Schüler können

Strukturierte Programmierung
(1) *Algorithmen* mit den Grundbausteinen *Anweisung*, *Bedingung*, *Schleife* und *Verzweigung* sowie unter Verwendung von *Variablen* in einer geeigneten textuellen Programmiersprache implementieren
P 2.2 Modellieren und Implementieren 9 F IMP 3.2.1.2 Algorithmen (1) F INF 3.1.2 Algorithmen (1) F INF7 3.1.2 Algorithmen (1), (5)
(2) Logische Verknüpfungen (UND, ODER, NICHT) in Bedingungen von *Schleifen* und *Verzweigungen* verwenden
I 3.1.3 Rechner und Netze (7)
(3) die Begriffe *Variable*, *Bezeichner*, *Datentyp*, *Wert*, *Deklaration*, *Initialisierung*, *Wertzuweisung* und *Gültigkeitsbereich* erläutern
P 2.1 Strukturieren und Vernetzen 2 F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (1)
(4) *primitive Datentypen* für ganzzahlige Werte, Gleitkommawerte und boolesche Werte unter Beachtung ihres *Wertebereichs* verwenden
P 2.1 Strukturieren und Vernetzen 2 I 3.1.1 Daten und Codierung (1) F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (2)
(5) implizite und explizite *Typumwandlungen* (*type casts*) zur Konvertierung von Werten unterschiedlicher *Datentypen* ineinander verwenden und dabei auftretende Probleme beschreiben
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (3) F INF 3.3.2.1 Strukturierte und objektorientierte Programmierung (3)
(6) Operationen auf *Zeichenketten* durchführen (unter anderem Verketten)
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (4)
(7) Zufallszahlen in eigenen Programmen verwenden
F IMP 3.2.1.2 Algorithmen (3) F INF 3.1.2 Algorithmen (4) F M 3.3.5 Leitidee Daten und Zufall
(8) *Unterprogramme* – auch mit *Parametern* und *Rückgabewerten* – sinnvoll verwenden
P 2.1 Strukturieren und Vernetzen 2, 6, 7 P 2.2 Modellieren und Implementieren 6 F IMP 3.2.1.2 Algorithmen (6) F INF 3.1.2 Algorithmen (5) L PG Selbstregulation und Lernen
(9) den Unterschied zwischen verschiedenen Fehlerarten (*Compilerfehler*/*Laufzeitfehler* und *syntaktisch*/*semantisch*) erläutern
F D 3.3.2.1 Struktur von Äußerungen F IMP 3.2.1.2 Algorithmen (8), (9) F INF 3.1.2 Algorithmen (6)
(10) vorgegebenen Code auf seine Funktionsweise hin analysieren und dessen Wirkung beschreiben
P 2.1 Strukturieren und Vernetzen 3 P 2.2 Modellieren und Implementieren 4 P 2.3 Kommunizieren und Kooperieren 4 P 2.4 Analysieren und Bewerten 1 F IMP 3.2.1.2 Algorithmen (11) F INF 3.1.2 Algorithmen (7) L MB Medienanalyse
(11) Algorithmen in den Notationsformen *Nassi-Shneiderman-Diagramm* und *Pseudocode* 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 F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (19)
(12) Anpassungen zu vorgegebenem Code implementieren
P 2.1 Strukturieren und Vernetzen 3 P 2.2 Modellieren und Implementieren 4, 6, 9, 10 P 2.3 Kommunizieren und Kooperieren 4 P 2.4 Analysieren und Bewerten 4 F IMP 3.2.1.2 Algorithmen (12) F INF 3.1.2 Algorithmen (8)
(13) vorgegebene Programmbibliotheken für eigene Programme sinnvoll verwenden
P 2.3 Kommunizieren und Kooperieren 4 F INF 3.1.2 Algorithmen (9) L PG Selbstregulation und Lernen
(14) *Algorithmen* entwerfen und implementieren, die Benutzereingaben anfordern und auswerten
P 2.1 Strukturieren und Vernetzen 5, 7 P 2.2 Modellieren und Implementieren 9 F IMP 3.2.1.2 Algorithmen (16) F INF 3.1.2 Algorithmen (10)
(15) Programmcode implementieren, der Daten aus Dateien einliest, verarbeitet und in Dateien schreibt (mittels geeigneter Bibliotheken)
P 2.1 Strukturieren und Vernetzen 7 P 2.3 Kommunizieren und Kooperieren 4 F IMP 3.2.1.2 Algorithmen (17) F INF 3.1.2 Algorithmen (11)
(16) *Arrays* zur Speicherung und Verarbeitung von Daten verwenden
F IMP 3.2.1.2 Algorithmen (4) F INF 3.1.2 Algorithmen (12)
(17) grundlegende *Algorithmen* auf *Arrays* (zum Beispiel Füllen mit Werten, Maximumsuche, Summenbildung, Bubblesort) erläutern und implementieren
F IMP 3.2.1.2 Algorithmen (5) F INF 3.1.2 Algorithmen (13)
(18) elementare vergleichsbasierte Sortierverfahren (*Bubblesort*, *Selectionsort*, *Insertionsort*) beschreiben, händisch durchführen und eines davon implementieren

P 2.2 Modellieren und Implementieren 9, 11
P 2.4 Analysieren und Bewerten 3
F INF 3.2.2.2 Algorithmen auf Datenstrukturen (5)

(19) Strategien (zum Beispiel Debugger, schrittweise Ausführung, Logging) anwenden, um das Verhalten von Programmcode zur *Laufzeit* zu beobachten

P 2.2 Modellieren und Implementieren 12
P 2.4 Analysieren und Bewerten 1
F IMP 3.2.1.2 Algorithmen (14)
F INF 3.1.2 Algorithmen (14)
L PG Selbstregulation und Lernen

L MB Informationstechnische Grundlagen; Produktion und Präsentation

Objektorientierte Programmierung

(20) eigene *Klassendefinitionen* mit *Attributen* und *Methoden* implementieren

P 2.1 Strukturieren und Vernetzen 2, 7
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (5)

(21) *Referenzvariablen*/*Referenztypen* mit *primitiven Variablen* / *primitiven Datentypen* vergleichen

F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (6)

(22) *Methoden* mit und ohne *Rückgabewert* sowie mit und ohne *Parameter* implementieren und den Begriff der *Methodensignatur* 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
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (8)

(23) die Funktion von *Zugriffsmodifikatoren* (*public*, *private*, *protected*) erläutern und diese verwenden

F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (9)

(24) die *Kapselung von Attributen* und *Methoden* (*Geheimnisprinzip*) erläutern und in eigenen Implementierungen verwenden (unter anderem Zugriff auf Attribute über *Zugriffsmethoden*)

P 2.1 Strukturieren und Vernetzen 7
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (10)

(25) die Rolle von *Konstruktoren* beim Erzeugen von *Instanzen einer Klasse* (*Objekten*) erläutern und diese implementieren

P 2.1 Strukturieren und Vernetzen 7
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (13)

(26) das Konzept der *Vererbung* erläutern und in Modellierungen und Implementierungen anwenden

P 2.1 Strukturieren und Vernetzen 3
P 2.2 Modellieren und Implementieren 4, 9
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (16)

(27) *UML-Klassendiagramme* mit *Vererbung* und *Assoziation* 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 3.1.1 Daten und Codierung (11)
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (20)

(28) 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
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (21)
L MB Kommunikation und Kooperation
L PG Selbstregulation und Lernen

(29) Dokumentationen zu gegebenem Quellcode und Bibliotheken nutzen

P 2.2 Modellieren und Implementieren 1, 6, 10
P 2.3 Kommunizieren und Kooperieren 4
F INF 3.2.2.1 Strukturierte und objektorientierte Programmierung (22)
L MB Kommunikation und Kooperation
L PG Selbstregulation und Lernen

(30) ein interaktives Programm (zum Beispiel App, Webanwendung, Desktopanwendung) mit einer einfachen *grafischen Benutzerschnittstelle* (zum Beispiel mit Buttons, Texteingabe und Ausgabe) implementieren, je nach Sprache unter Verwendung geeigneter (didaktischer) Toolkits und/oder GUI-Builder

P 2.1 Strukturieren und Vernetzen 9
P 2.2 Modellieren und Implementieren 6, 10, 12
F IMP 3.3.1.2 Algorithmen (3)
F INF 3.1.2 Algorithmen (15)
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; Produktion und Präsentation

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 Schülern entwerfen und untersuchen einfache Schaltnetze und erhalten anhand des Aufbaus von Halb- und Volladdierer ein Einblick in die Verarbeitung von Daten auf Hardwareebene.

Die Schülerinnen und Schüler können

Rechnernetze und Routing
(1) den grundlegenden Aufbau eines *lokalen Rechnernetzes* und die Rolle seiner Komponenten (Endgerät, Verbindung, Verteiler) erklären
F IMP 3.1.1.3 Rechner und Netze (1) F INF 3.1.3 Rechner und Netze (1)
(2) die Notwendigkeit einer eindeutigen *Adressierung* zur Kommunikation in Netzen erläutern und hierfür Beispiele nennen (*IP-Adresse* und zum Beispiel Handynummer, E-Mail-Adresse)
F IMP 3.1.1.3 Rechner und Netze (2) F INF 3.1.3 Rechner und Netze (2)
(3) das Prinzip der *Namensauflösung* (*DNS* und zum Beispiel Kontaktliste, Telefonbuch) erläutern
I 3.1.3 Rechner und Netze (3) F IMP 3.1.1.3 Rechner und Netze (5)
(4) ein *lokales Rechnernetz* mit *DNS* und *Webserver* 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) F INF 3.1.3 Rechner und Netze (4)
(5) Schemata beschreiben, mit denen eine Unterscheidung von *Adressen* 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) F INF 3.1.3 Rechner und Netze (5)
(6) das Problem des *Routings* zwischen Netzen erläutern und in einer geeigneten Simulationsumgebung ein Routingszenario durchführen
F IMP 3.3.1.3 Rechner und Netze (8) F INF 3.1.3 Rechner und Netze (6)
L MB Informationstechnische Grundlagen; Kommunikation und Kooperation; Mediengesellschaft
Schaltnetze
(7) aus *logischen Gattern* (unter anderem *AND*, *OR*, *XOR*, *NOT*) Schaltnetze entwerfen, diese untersuchen und ihre *Wahrheitstafeln* ermitteln
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 I 3.1.2 Algorithmen (2) F IMP 3.3.1.3 Rechner und Netze (1) F INF 3.3.3 Rechner und Netze (1)
(8) Aufbau und Funktion von *Halbaddierer* und *Volladdierer* beschreiben und daraus in einer Simulationsumgebung einen Mehrbitaddierer erstellen
P 2.1 Strukturieren und Vernetzen 6, 8 P 2.2 Modellieren und Implementieren 7 P 2.3 Kommunizieren und Kooperieren 1 F IMP 3.3.1.3 Rechner und Netze (5) F INF 3.3.3 Rechner und Netze (2)
L MB Informationstechnische Grundlagen

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 moderne Verfahren.
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 können

(1) das *Vigenère-Verfahren* erklären und durchführen
F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (1) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (1)
(2) *monoalphabetische Substitution* und *polyalphabetische Substitution* vergleichen
P 2.4 Analysieren und Bewerten 3 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (2) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (2)
(3) eine grundlegende Angriffsstrategie auf das *Vigenère-Verfahren* erklären und an einfachen Beispielen durchführen
F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (3) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (3)
(4) das *One-Time-Pad*-Verfahren 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) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (4)
(5) erläutern, dass die Sicherheit von Verschlüsselungsverfahren nicht von der Geheimhaltung des *Algorithmus* abhängen darf (*Kerckhoffs'sches Prinzip*)
P 2.4 Analysieren und Bewerten 7 F IMP 3.1.1.4 Informationsgesellschaft und Datensicherheit (6) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (5)
(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) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (6) 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) das Konzept der *asymmetrischen Verschlüsselung* (*privater/öffentlicher Schlüssel*) erklären
F IMP 3.3.1.4 Informationsgesellschaft und Datensicherheit (1) F INF 3.1.4 Informationsgesellschaft und Datensicherheit (9)
(8) *asymmetrische* und *symmetrische Verschlüsselung* vergleichen (Schlüsselverwaltung, Schlüsseltausch, Geschwindigkeit)
P 2.2 Modellieren und Implementieren 1 P 2.4 Analysieren und Bewerten 3 F INF 3.2.4 Informationsgesellschaft und Datensicherheit (1)
(9) Anwendungsbereiche von *Public-Key-Infrastrukturen* beschreiben (zum Beispiel digitaler Personalausweis, Transportverschlüsselung, Mailverschlüsselung) und im Hinblick auf Sicherheitsaspekte bewerten
P 2.3 Kommunizieren und Kooperieren 9, 10 P 2.4 Analysieren und Bewerten 5, 8 F INF 3.2.4 Informationsgesellschaft und Datensicherheit (6)
(10) Maßnahmen zur Gewährleistung von *Datensicherheit* und *Datenschutz* beim Einsatz von Informatiksystemen erläutern
P 2.4 Analysieren und Bewerten 1, 3, 4, 7, 8, 9 F INF 3.2.4 Informationsgesellschaft und Datensicherheit (9) L BTV Minderheitenschutz; Wertorientiertes Handeln
(11) Szenarien bewerten, in denen Daten massenweise erhoben, gespeichert und weiterverarbeitet werden
P 2.4 Analysieren und Bewerten 5, 8, 9 F INF 3.2.4 Informationsgesellschaft und Datensicherheit (10) 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 Chancen und Risiken der Lebensführung; Verbraucherrechte