

## **6. Jahrgangsstufe: NT6 2 Schwerpunkt Informatik (ca. 28 Std.)**

### **NT6 2.1 Informationsdarstellung mit Grafik-, Text- und Multimediadokumenten (ca. 18Std.)**

#### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- interpretieren, vergleichen und bewerten verschiedene Darstellungen von Informationen.
- analysieren Grafik-, Text- und Multimediadokumente aus objektorientierter Sicht. Damit entwickelt sich das Verständnis, dass die Dokumente Informationseinheiten (z. B. Absätze bei Textdokumenten, Folien bei Multimediadokumenten) enthalten, die durch spezifische Eigenschaften charakterisiert sind und die miteinander in Beziehung stehen. So erhalten die Schülerinnen und Schüler einerseits Einblick in ein für die Informatik zentrales Modellierungskonzept; andererseits wird ihr Verständnis vertieft, dass diese Dokumentarten jeweils eine typische Grundstruktur haben, was die Nutzung entsprechender Software unabhängig von einem speziellen Produkt erleichtert.
- beschreiben Objekte (Informationseinheiten) durch ihre Eigenschaften sowie Modifikationen an diesen Objekten (insbesondere Attributwertänderungen mithilfe von Methoden). Hierbei verwenden sie eine einfache, einheitliche und intuitiv verständliche Beschreibungssprache in Form der Punktnotation (einer typischen Notationsform in der Informatik).
- abstrahieren Informationseinheiten gleicher Art zu Klassen, interpretieren diese als Bauplan für Objekte und erweitern damit ihre Fertigkeiten im objektorientierten Modellieren.
- stellen Struktur und Beziehungen der Informationseinheiten durch Objekt- und Klassendiagramme sachgerecht dar. Damit steht ihnen eine übersichtliche grafische Darstellungsmöglichkeit zur Verfügung, die auch in vielen anderen Bereichen der Informatik verwendet wird.
- nutzen situationsgerecht Grafik-, Text- und Präsentationsprogramme zur Gestaltung einfacher Dokumente, um Informationen aussagekräftig darzustellen.
- beurteilen anhand von Praxisbeispielen (z. B. Foto, Grundrissplan), ob sich für die Darstellung einer Information abhängig vom Einsatzzweck eine Vektor- oder Pixelgrafik besser eignet.

### **Inhalte zu den Kompetenzen:**

- Darstellungsformen von Information, z. B. Text, Bild, Diagramm, Ton, Pantomime, Zeichnung, Skizze
- allgemeine Aspekte der Analyse bzw. Modellierung: Objekt, Attribut, Methode, Beziehung, Klasse
- Analyse bzw. Modellierung von Vektorgrafikdokumenten, u.a. mithilfe der Klassen GRAFIKDOKUMENT, TEXTFELD, RECHTECK oder QUADRAT, ELLIPSE oder KREIS, LINIE und der Enthält-Beziehung
- Analyse bzw. Modellierung von Textdokumenten, u.a. mithilfe der Klassen TEXTDOKUMENT, ABSATZ und ZEICHEN und der Enthält-Beziehung
- Analyse bzw. Modellierung von Multimediadokumenten, u.a. mithilfe der Klassen MULTIMEDIADOKUMENT, FOLIE und z. B. BILD oder AUDIO und der Enthält-Beziehung; ggf. Animieren von Objekten mithilfe entsprechender Methoden
- Objekt- und Klassenkarte, Objekt- und Klassendiagramm, Punktnotation
- Vektor- und Pixelgrafik als unterschiedliche Grafikarten
- grundlegende Funktionen von Standardsoftware (z. B. Speichern, Öffnen, Schließen, Kopieren, Einfügen)
- Verbesserung der Informationsdarstellung durch geeignetes Ändern von Attributwerten (z. B. Textformatierung)
- spezielle Aspekte der Softwarebeherrschung bei Vektorgrafiksoftware (z. B. Gruppieren, Anordnung der Ebenen)
- spezielle Aspekte der Softwarebeherrschung bei Textverarbeitungssoftware (z. B. Einfügen von Bildern, Erstellen von Tabellen)
- spezielle Aspekte der Softwarebeherrschung bei Präsentationsprogrammen: Bearbeitungs- und Vorführmodus, ggf. Animation
- Fachbegriffe: Attribut, Methode, Klasse, Beziehung, Punktnotation, Objektdiagramm, Klassendiagramm

## **NT6 2.2 Projekt: Erstellen einer Multimediapräsentation (ca. 5Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- erstellen innerhalb eines vorgegebenen Zeitrahmens eine ansprechende Multimediapräsentation zu einem vorgegebenen Thema und berücksichtigen dabei sinnvolle Kriterien für die Qualität einer Präsentationsgestaltung.
- beachten bei der Zusammenstellung der Inhalte für die Multimediapräsentation grundlegende Vorgaben des Urheberrechts.

### **Inhalte zu den Kompetenzen:**

- Kriterien für die Qualität einer Multimediapräsentation, z. B. Textanteil je Folie, Schriftgröße, Farbwahl, zielgerichtete Auswahl von Animationen
- Urheberrecht im Kontext der Erstellung von schulischen Arbeiten, Quellenangabe

## **NT6 2.3 Hierarchische Informationsstrukturen – Dateisystem (ca. 5Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- ordnen Informationen aus geeigneten einfachen Beispielen ihrer Erfahrungswelt hierarchisch (z. B. Stammbaum).
- stellen baumartige Informationsstrukturen mithilfe entsprechender Baumdiagramme dar.
- analysieren die in einem Dateisystem abgelegte Anordnung von Dateien und Ordnern, erkennen die zugrunde liegende hierarchische Struktur und stellen diese in Objektdiagrammen und abstrahiert als Klassendiagramm dar.
- nutzen einen Dateimanager, um Dateien und Ordner in einer hierarchischen Struktur geeignet zu organisieren. Damit können sie Dateien sicher abspeichern und wiederfinden.
- geben die Lage von Dateien, die auf einem Rechner gespeichert sind, durch Pfade an. Damit beherrschen sie eine grundlegende Technik für den Zugriff auf Dateien, wie sie beispielsweise in Netzwerken eingesetzt wird.

### **Inhalte zu den Kompetenzen:**

- Modellierung der hierarchischen Struktur im Dateisystem, u.a. mit den Klassen DATEI und ORDNER und der rekursiven Enthält-Beziehung der Klasse ORDNER
- Baum als Möglichkeit der Darstellung bestimmter hierarchischer Strukturen: Wurzel, Knoten, Kante, Blatt, Pfad
- Fachbegriffe: Ordner, Datei, Pfad, Baum, Wurzel, Knoten, Kante, Blatt



## **7. Jahrgangsstufe: 2 Schwerpunkt Informatik (ca. 28Std.)**

### **NT7 2.1 Vernetzte Informationsstrukturen (ca. 8Std.)**

#### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- analysieren Strukturen vernetzter Informationen aus ihrem Lebensumfeld (z. B. Lexikoneinträge, Webseiten der Schule) und modellieren entsprechende Hypertextstrukturen mithilfe objektorientierter Konzepte.
- nutzen effektive Strategien zur Informationsbeschaffung im Internet und bewerten ihre Suchergebnisse kritisch hinsichtlich Wahrheitsgehalt und Qualität, indem sie z. B. die Art der Informationsquellen (etwa Werbeseiten, Blogeinträge, wissenschaftliche Publikationen) berücksichtigen.
- nutzen die grundlegenden Funktionen eines geeigneten Werkzeugs (WYSIWYG-Editor) zur Erstellung einer Hypertextstruktur; hierbei berücksichtigen sie grundlegende Aspekte des Urheberrechts und wenden einfache Zitiernormen (Quellenangaben) an.
- beschreiben den prinzipiellen Mechanismus für die Übermittlung elektronischer Dokumente unter Verwendung ihres Wissens über Struktur und Funktionsweise des Internets.

#### **Inhalte zu den Kompetenzen:**

- Analyse und Modellierung von Hypertextstrukturen, u.a. mithilfe der Klasse VERWEIS; Zieladresse als Attribut eines Verweises
- Darstellung der Objektstruktur einer Hypertextstruktur: Graph, bestehend aus Knoten und Kanten
- Struktur und Funktionsweise des Internets: Client, Server, Vermittlungsrechner (Router); Dienste (u.a. World Wide Web); einfache Beispiele für die Adressierung im Internet (z. B. IP-Adresse, URL)
- Informationsquellen im Internet, z. B. Suchmaschinen, Enzyklopädien
- Fachbegriffe: Hypertext, Verweis, Graph, Client, Server

## **NT7 2.2 Chancen und Risiken digitaler Kommunikation (ca. 5Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- bewerten verschiedene digitale Kommunikationsmöglichkeiten hinsichtlich ihrer Chancen, Risiken und Auswirkungen auf das Individuum und die Gesellschaft, um sie im privaten oder schulischen Kontext sicher sowie aus medienpädagogischer Sicht sinnvoll einsetzen zu können.
- beachten grundlegende Regeln zum Schutz von Persönlichkeitsrechten im Rahmen der digitalen Kommunikation. Dabei wird ihnen das Spannungsfeld zwischen dem Recht des Einzelnen und den Interessen anderer bewusst.

### **Inhalte zu den Kompetenzen:**

- digitale Kommunikationsmöglichkeiten, z. B. E-Mail, Messenger, soziale Netzwerke
- Schutz eigener Endgeräte durch Software (z. B. Firewall und Virenschutz) und durch Verwendung sicherer Passwörter
- Persönlichkeitsrecht und Datenschutz im Rahmen der digitalen Kommunikation, u.a. personenbezogene Daten, Recht am eigenen Bild
- Gefahrenpotenziale (z. B. Viren, Trojaner, gefälschte virtuelle Identität, Abofallen, Fake News), Verhaltensregeln im Kontext der digitalen Kommunikation (auch unter Berücksichtigung ethischer Aspekte), Cybermobbing als Beispiel eines Missbrauchs digitaler Kommunikationsmittel

## **NT7 2.3 Beschreibung von Abläufen durch Algorithmen (ca. 11Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- analysieren und strukturieren geeignete Problemstellungen u.a. aus ihrer Erfahrungswelt (z. B. Bedienung eines Geräts), entwickeln Algorithmen zu deren Lösung und beschreiben diese unter effizienter Verwendung von Kontrollstrukturen.
- setzen unter sinnvoller Nutzung algorithmischer Bausteine einfache Algorithmen mithilfe geeigneter Programmierwerkzeuge um.

**Inhalte zu den Kompetenzen:**

- Algorithmus: Definition des Begriffs, Strukturelemente (Anweisung, Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholung mit fester Anzahl, Wiederholung mit Bedingung)
- Fachbegriffe: Algorithmus, Anweisung, Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholung mit fester Anzahl, Wiederholung mit Bedingung

**NT7 2.4 Projekt (ca. 4Std.)****Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- entwickeln innerhalb eines festen Zeitrahmens im Team eine Hypertextstruktur zu einem vorgegebenen Thema oder eine Lösung zu einer geeigneten Aufgabenstellung aus der Algorithmik (z. B. Physical Computing).
- nutzen ihre Informatikkenntnisse zur Sicherung von Zwischenständen und ggf. zur Dokumentation ihres Arbeitsprozesses.

**Inhalte zu den Kompetenzen:**

- Grundlagen der Projektarbeit, insbesondere Zeitmanagement und Arbeitsteilung, ggf. Dokumentation





## 9. Jahrgangsstufe

### Inf9 Lernbereich 1: Funktionen und Datenflüsse, Tabellenkalkulationsprogramm (ca. 14Std.)

#### Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- abstrahieren Daten verarbeitende Prozesse mit mehreren Eingaben und einer Ausgabe zu Funktionen.
- modellieren die durch Funktionen ausgelösten Datenflüsse mithilfe von Datenflussdiagrammen.
- entwickeln neue Funktionen durch Verkettung gegebener Funktionen. Sie wenden damit ein grundlegendes Konzept der funktionalen Modellierung an.
- setzen zur automatisierten Datenverarbeitung Datenflussdiagramme und Funktionen in Formeln eines Tabellenkalkulationssystems um und überprüfen durch geeignete Eingaben Modell und Umsetzung.
- lösen praxisnahe Aufgabenstellungen, beispielsweise aus dem kaufmännischen Bereich oder der Mathematik, sachgerecht durch Anwendung der funktionalen Sichtweise, realisieren ihre Lösung mit einem Tabellenkalkulationsprogramm und bewerten deren Qualität. Dabei nutzen sie grundlegende Möglichkeiten eines Tabellenkalkulationsprogramms, u.a. sinnvolle Nutzung von Adressierung und passende Gestaltung.

#### Inhalte zu den Kompetenzen:

- Tabellenkalkulationsprogramm: Tabellenblatt, Zelle, Formel, Funktion (auch vordefinierte Funktion), Zellbezug (relative und absolute Adressierung)
- Datenflussdiagramm: Repräsentation einer Funktion, Datenfluss, Ein- und Ausgabe, Verteiler
- Funktion: Interpretation als Daten verarbeitender Prozess, vordefinierte Funktionen (u.a. bedingte Funktion), Verkettung von Funktionen, Parameter
- Fachbegriffe: Formel, Zellbezug (relativ, absolut), Funktion, Datenflussdiagramm, Verteiler

## **Inf9 Lernbereich 2: Grundlagen der Datenmodellierung und relationaler Datenbanksysteme (ca. 10Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- analysieren Datenbestände einfacher Beispiele aus der Praxis (z. B. Mitgliederverzeichnis) und modellieren diese objektorientiert.
- überführen objektorientierte Datenmodelle in entsprechende Tabellenschemata und setzen diese in einem Datenbanksystem um.
- konzipieren geeignete SQL-Abfragen, um zielgerichtet Informationen aus einer Datenbanktabelle zu gewinnen.

### **Inhalte zu den Kompetenzen:**

- objektorientiertes Datenmodell: Objekt, Klasse, Attribut
- relationales Modell: Tabellenschema, Primärschlüssel, Datentyp
- relationales Datenbanksystem: Datenbank, Datenbankmanagementsystem, Tabelle, Datensatz
- Abfrage: Interpretation als Funktion, Ergebnistabelle als Ergebnis einer Abfrage
- Abfragesprache am Beispiel von SQL: select, from, where; Verknüpfung von Bedingungen
- Fachbegriffe: (relationales) Datenbanksystem, (relationale) Datenbank, Datenbankmanagementsystem, Tabellenschema, Datentyp, Datensatz, Abfrage, Primärschlüssel, Ergebnistabelle

## **Inf9 Lernbereich 3: Grundlagen der objektorientierten Modellierung und Programmierung (ca. 26Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- analysieren Objekte aus ihrer Erfahrungswelt (z. B. Fahrzeuge, Personen) hinsichtlich ihrer Eigenschaften (Attribute) und Fähigkeiten (Methoden) und abstrahieren sie zu Klassen. Sie stellen Objekte und Klassen als Grundlage einer möglichen Implementierung grafisch dar.
- deklarieren eine Klasse sowie die zugehörigen Attribute und Methoden in einer objektorientierten Programmiersprache.

- verwenden bei der Implementierung Wertzuweisungen, um Attributwerte zu ändern, und interpretieren diese als Zustandsänderung des zugehörigen Objekts.
- formulieren unter Verwendung der Kontrollstrukturen Algorithmen zu geeigneten Problemstellungen, u.a. durch grafische Darstellungen.
- implementieren Methoden auf der Grundlage gegebener Algorithmen objektorientiert, wobei sie sich des Unterschiedes zwischen Methodendefinition und Methodenaufruf bewusst sind. Dabei nutzen sie ggf. auch Methoden anderer Klassen.
- analysieren, interpretieren und modifizieren Algorithmen, wodurch sie die Fähigkeit erlangen, fremde Programme flexibel einzusetzen und kritisch zu bewerten.
- modellieren durch Klassendiagramme einfache Generalisierungshierarchien zu geeigneten Strukturen aus ihrer Erfahrungswelt.
- implementieren mithilfe einer objektorientierten Sprache einfache Generalisierungshierarchien; dabei nutzen sie das Konzept der Vererbung sowie die Möglichkeit, Methoden zu überschreiben.

### **Inhalte zu den Kompetenzen:**

- objektorientierte Konzepte, u.a. Objekt, Klasse, Attribut, Attributwert, Methode
- Variablenkonzept; Arten von Variablen: Parameter, lokale Variable und Attribute; Übergabewert
- Wertzuweisung zur Änderung von Variablenwerten
- Methoden: Methodenkopf, Methodenrumpf, Methodendefinition, Methodenaufruf, Übergabewert, Rückgabewert; Konstruktor als spezielle Methode; Standardmethoden zum Geben und Setzen von Attributwerten
- Algorithmus: Strukturelemente, grafische Darstellung, Pseudocode
- Datentypen: ganze Zahlen, Gleitkommazahlen, Wahrheitswerte, Zeichen, Zeichenketten
- Generalisierung und Spezialisierung: Ober- und Unterklasse, Vererbung von Attributen und Methoden an Unterklassen, Überschreiben von Methoden
- Fachbegriffe: Parameter, Übergabewert, Rückgabewert, lokale Variable, Wertzuweisung, Konstruktor, Methodenkopf, Methodenrumpf, Vererbung, Generalisierung, Spezialisierung, Oberklasse, Unterklasse

## **Inf9 Lernbereich 4: Datenschutz (ca. 6Std.)**

### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- bewerten Regelungen zum Datenschutz im Spannungsfeld zwischen den Persönlichkeitsrechten des Einzelnen und wirtschaftlichen sowie öffentlichen Interessen, beispielsweise bei der Fahndung nach Straftätern.
- nutzen das Internet verantwortungsvoll unter Berücksichtigung ihrer Kenntnisse über Möglichkeiten und Risiken dieses Mediums und reflektieren dabei, wodurch der Schutz persönlicher Daten erhöht und die Gefahr des Missbrauchs minimiert werden kann. Insbesondere wägen sie kriteriengeleitet ihren Umgang mit datenbankgestützten Diensten und Portalen ab. In diesem Kontext reflektieren sie den Einfluss digitaler Medien (z. B. Messenger-Dienste, soziale Netzwerke) auf sich und die Gesellschaft (Medienwirkung).
- bewerten Chancen und Risiken der automatisierten Analyse von großen Datenbeständen (Data-Mining), auch im Hinblick auf gesellschaftliche Auswirkungen.

### **Inhalte zu den Kompetenzen:**

- Datenschutzgesetze: Zweck, Grundsätze (z. B. Verbotprinzip mit Erlaubnisvorbehalt), Rechte von Betroffenen
- Datenschutz: Schutz personenbezogener Daten (insbesondere im Kontext der Mehrbenutzerproblematik bei Datenbanken), Datenmissbrauch, z. B. Identitätsdiebstahl
- Fachbegriffe: Datenschutz, Data-Mining

## **10. Jahrgangsstufe**

### **Inf10 Lernbereich 1: Datenmodellierung und relationale Datenbanksysteme (ca. 17Std.)**

Die Schülerinnen und Schüler ...

- analysieren und strukturieren mithilfe objektorientierter Konzepte Datenbestände geeigneter Beispiele aus der Praxis (z. B. Bibliotheksverwaltung) und stellen das daraus entwickelte Datenmodell als Klassendiagramm mit mehreren Klassen dar.
- überführen objektorientierte Datenmodelle in entsprechende relationale Modelle und setzen diese in einem Datenbanksystem um.
- konzipieren geeignete SQL-Abfragen, um zielgerichtet Informationen aus einer relationalen Datenbank zu gewinnen.
- erkennen Redundanzen und Anomalien in einer relationalen Datenbank und beurteilen die dadurch entstehende Problematik im Hinblick auf die Konsistenz des Datenbestands.
- bewerten Chancen und Risiken der automatisierten Analyse verknüpfter Datenbestände, auch im Hinblick auf gesellschaftliche Auswirkungen.

#### **Inhalte zu den Kompetenzen:**

- objektorientiertes Datenmodell: Objekt, Klasse, Attribut, Beziehung, Kardinalität
- relationales Modell: Tabellenschema, Datenbankschema, Primär- und Fremdschlüssel, Datentyp
- relationales Datenbanksystem
- Redundanz und Konsistenz von Datenbeständen, Anomalien
- Abfragesprache am Beispiel von SQL: Verknüpfung von Bedingungen; Abfrage über verknüpfte Tabellen
- Fachbegriffe: Datenbankschema, Primär- und Fremdschlüssel, Kardinalität, Redundanz, Konsistenz, Anomalie

### **Inf10 Lernbereich 2: Objektorientierte Modellierung und Programmierung (ca. 27Std.)**

#### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- wenden eine eindimensional indizierte Datenstruktur bei einfachen Problemstellungen zur Speicherung und Verwaltung gleichartiger Daten an.
- analysieren und modellieren Objektbeziehungen und das Kommunikationsverhalten innerhalb eines Systems (beispielsweise eines Fahrradverleihs).
- implementieren die im Klassendiagramm festgelegten Beziehungen sachgerecht durch Referenzen, um während der Laufzeit des Programms die Kommunikation zwischen den entsprechenden Objekten durch den Aufruf geeigneter Methoden zu ermöglichen. Dabei wird ihnen die Grundidee der Datenkapselung bewusst.
- modellieren und implementieren im Rahmen eines einfachen Beispiels aus der Praxis (etwa der Mitgliederverwaltung eines Vereins) eine Struktur zur Verwaltung gleichartiger Objekte. Hierbei entwickeln sie ausgewählte Methoden, z. B. zum Einfügen, Entfernen und Bearbeiten von Elementen.
- nutzen in Generalisierungshierarchien das Konzept des Polymorphismus zur Implementierung verschiedener Verhaltensweisen, z. B. die spartenabhängige Berechnung der Mitgliedsbeiträge eines Sportvereins.

#### **Inhalte zu den Kompetenzen:**

- eindimensional indizierte Datenstruktur (Array/Feld): Index, Element, Länge
- Interpretation von Klassen als Datentypen
- Umsetzung von Klassenbeziehungen unterschiedlicher Kardinalitäten mithilfe von Referenzen
- Kommunikation zwischen Objekten durch Methodenaufrufe, Datenkapselung durch kontrollierten Zugriff auf die Attribute
- Polymorphismus und Überschreiben von Methoden
- Fachbegriffe: Referenz, Kardinalität, Array/Feld, Index, Datenkapselung, Polymorphismus

### **Inf10 Lernbereich 3: Projekt (ca. 12Std.)**

#### **Kompetenzerwartungen**

Die Schülerinnen und Schüler ...

- erstellen im Team einen Projektplan, um eine Datenbank bzw. ein objektorientiertes Programm zu einem Szenario aus ihrer Erfahrungswelt, wie z. B. einem Buchungssystem (ggf. mit Benutzerschnittstelle) oder einem einfachen Spiel, zu entwickeln.

- analysieren das gegebene Szenario und modellieren relevante Ausschnitte durch ein Klassendiagramm und ggf. weitere Diagramme. Dabei arbeiten sie ggf. mit bereits existierenden Modellen bzw. Modellausschnitten.
- implementieren zum gegebenen Szenario arbeitsteilig und auf Grundlage der vorhandenen Modellierung die Datenbank einschließlich geeigneter Abfragen bzw. das objektorientierte Programm.
- testen die entwickelte Datenbank bzw. das entwickelte Programm hinsichtlich der Anforderungen des gegebenen Szenarios.
- dokumentieren ihre Projektarbeit geeignet.

**Inhalte zu den Kompetenzen:**

- Datenbank- bzw. Softwareentwicklungsprojekt: Planung, Modellierung, Implementierung, Test, Dokumentation





## Jahrgangsstufe 11

Information und Kommunikation sind zentrale Begriffe der Informatik. In den Jahrgangsstufen 9 und 10 haben sich die Schüler insbesondere mit der Analyse sowie mit der für eine maschinelle Verarbeitung geeigneten Darstellung von Information beschäftigt und dabei verschiedene Techniken der Modellierung kennengelernt. Darauf aufbauend lernen sie in Jahrgangsstufe 11 neue Konzepte anzuwenden, die es ihnen erlauben, größere Systeme effizienter zu modellieren. Sie greifen nun auf rekursive Datenstrukturen zurück und erkennen deren Nutzen als häufig verwendbare Modellierungsmuster. Bei der Erstellung größerer Softwareprodukte eignen sich die Schüler neben effizienten Designstrategien hilfreiche Vorgehensweisen für die Arbeit in einem Team an, die auch außerhalb informatischer Aufgabenstellungen gewinnbringend einsetzbar sind.

Um die Möglichkeiten der Kommunikation zwischen Mensch und Maschine besser beurteilen zu können, betrachten die Schüler in Jahrgangsstufe 12 das dabei verwendete Hilfsmittel Sprache. Die Erkenntnis, dass für die Kommunikation mit einer Maschine exakte Vereinbarungen unentbehrlich sind, führt sie zum Begriff der formalen Sprache. Bei deren praktischer Anwendung wird den Jugendlichen bewusst, dass der Kommunikation Mensch-Maschine durch den nötigen Formalismus große Beschränkungen auferlegt sind.

Beim weltweiten Austausch von Information spielt die Kommunikation zwischen vernetzten Rechnern eine entscheidende Rolle. Die Schüler lernen, dass es hierzu fester Regeln für das Format der auszutauschenden Daten sowie für den Ablauf des Kommunikationsvorgangs bedarf. Der gemeinsame Zugriff auf Ressourcen führt sie zum Problem der Kommunikation und Synchronisation parallel ablaufender Prozesse, bei dessen Lösung die Jugendlichen erneut den Anwendungsbereich ihrer Modellierungstechniken erweitern.

Grundlegende Kenntnisse über den Aufbau eines Rechners und seiner prinzipiellen Funktionsweise helfen den Schülern, den Kommunikationsvorgang mit einer Maschine besser zu verstehen. Die prinzipielle Automatisierbarkeit des Übersetzungsvorgangs von einer höheren Programmiersprache in eine Maschinensprache wird ihnen bei der Umsetzung einfacher Algorithmen mit einer maschinennahen Sprache deutlich.

Ein wichtiges Maß für die Realisierbarkeit von Algorithmen ist die Effizienz hinsichtlich des Zeitbedarfs. Bei der Untersuchung des Laufzeitverhaltens ausgewählter Algorithmen erkennen die Jugendlichen praktische Grenzen der Berechenbarkeit. Daneben gewinnen sie auch Einblicke in theoretische Grenzen der Berechenbarkeit, sodass sie die Einsatzmöglichkeiten automatischer Informationsverarbeitung realistischer einschätzen können.

Für interessierte Jugendliche bietet sich die Möglichkeit, Informatik auch als Seminar zu wählen.

## Inf 11.1 Rekursive Datenstrukturen

In Jahrgangsstufe 10 haben die Schüler im Rahmen der objektorientierten Modellierung schwerpunktmäßig am Erstellen geeigneter Methoden von Objekten sowie am Erkennen und Umsetzen von Objektbeziehungen gearbeitet. Verschiedenartige Aufgabenstellungen aus der Praxis verdeutlichen den Schülern nun, dass immer wieder Strukturen von Daten auftreten, die mit den bisher bekannten Datentypen nicht effizient beschrieben werden können. Die Schüler beschäftigen sich daher eingehend mit typischen rekursiven Datentypen, lernen deren Vorteile und breites Spektrum von Einsatzmöglichkeiten kennen und wenden sie an; dabei nutzen sie erstmals rekursive Methoden.

### Inf 11.1.1 Listen (ca. 29 Std.)

Die Schüler untersuchen die grundlegenden Eigenschaften der Datenstruktur Schlange, deren grundsätzlichen Aufbau sie bereits aus ihrem Alltag, z. B. von Warteschlangen, kennen. Eine erste Implementierung mit einem Feld zeigt schnell die Grenzen dieser statischen Lösung auf und führt die Jugendlichen zu einer dynamischen Datenstruktur wie der einfach verketteten Liste. Sie erarbeiten deren prinzipielle Funktionsweise sowie deren rekursiven Aufbau und wenden hierbei das Prinzip der Referenz auf Objekte an. Die Jugendlichen erkennen, dass die rekursive Struktur der Liste für viele ihrer Methoden einen rekursiven Algorithmus nahelegt. Sie verstehen, dass eine universelle Verwendbarkeit der Klasse Liste nur möglich ist, wenn auf eine klare Trennung von Struktur und Daten geachtet wird. An einfachen Beispielen aus der Praxis und deren Implementierung vertiefen die Schüler ihr Wissen und erfahren die flexible Verwendbarkeit dieses Datentyps.

- Methoden der Datenstruktur Schlange: Anfügen am Ende, Entfernen am Anfang
- allgemeines Prinzip und rekursive Struktur einer einfach verketteten Liste; graphische Veranschaulichung der Methoden zum Einfügen (auch an beliebiger Stelle), Suchen und Löschen
- rekursive Abläufe: rekursiver Methodenaufruf, Abbruchbedingung, Aufrufsequenz
- Implementierung einer einfach verketteten Liste als Klasse mittels Referenzen unter Verwendung eines geeigneten Softwaremusters (Composite); Realisierung der Methoden zum Einfügen, Suchen und Löschen
- Einsatz der allgemeinen Datenstruktur Liste bei der Bearbeitung eines Beispiels aus der Praxis: Verwaltung von Elementen verschiedener Datentypen mittels Vererbung
- Stapel und Schlange als spezielle Formen der allgemeinen Datenstruktur Liste

## **Inf 11.1.2 Bäume als spezielle Graphen (ca. 29 Std.)**

In Erweiterung ihrer Grundkenntnisse über das hierarchische Ordnungsprinzip lernen die Schüler den Baum als effiziente dynamische Datenstruktur kennen. Sie stellen fest, dass damit viele Strukturen aus anderen Gebieten abgebildet werden können. Am Beispiel der Suche in umfangreichen Datenbeständen wird den Jugendlichen deutlich, dass sich auch hier sehr oft Baumstrukturen einsetzen lassen, um die Effizienz der Informationsverarbeitung zu steigern. Bei der sprachlichen bzw. graphischen Darstellung und insbesondere bei der Implementierung dieser Datenstruktur vertiefen sie ihr Verständnis für das Prinzip der Rekursion. Im Rahmen praktischer Fragestellungen, z. B. zur Planung von Verkehrsrouten, wenden die Schüler auch die Datenstruktur Graph als Erweiterung der Struktur Baum an.

- allgemeines Prinzip und Struktur eines Baums (insbesondere Wurzel, Knoten, Kanten, Blatt) und des Spezialfalls geordneter Binärbaum
- Veranschaulichung und Implementierung der Methoden zum Einfügen und Suchen von Elementen in einem geordneten Binärbaum unter Verwendung der Rekursion
- Verfahren zur Auflistung aller Elemente eines geordneten Binärbaums: Preorder-, Inorder- und Postorder-Durchlauf; Realisierung der Methode zum Ausgeben mithilfe eines dieser Verfahren
- Implementierung der Klasse geordneter Binärbaum mit einer geeigneten Programmiersprache; Verwenden und Testen der Methoden an einem Anwendungsbeispiel (z. B. erweiterbares Wörterbuch als Suchbaum)
- die Datenstruktur Graph als Verallgemeinerung des Baums; Eigenschaften (gerichtet/ungerichtet, bewertet/unbewertet); Adjazenzmatrix
- Algorithmus zum Graphendurchlauf (z. B. Tiefensuche) bei einer Aufgabenstellung aus der Praxis

## **Inf 11.2 Softwaretechnik (ca. 26 Std.)**

Das Arbeiten in Projekten ist die typische Vorgehensweise bei der Entwicklung großer Systeme. Mit den bisher erworbenen Kenntnissen und Fertigkeiten sind die Schüler nun in der Lage, größere Softwaresysteme (z. B. Geschäftsabläufe einer Bank, Autovermietung) eigenständig zu gestalten. Hierbei bauen sie auch ihre Fähigkeit zur Planung und Durchführung von Projekten aus. Die Jugendlichen übernehmen verstärkt persönliche Verantwortung und erfahren die Notwendigkeit, eigene Ansichten und Ideen vor anderen darstellen und vertreten zu können. Sie setzen alle bisher erlernten Beschreibungstechniken der Informatik ein und machen sich damit deren Zusammenwirken in einem größeren Kontext bewusst. Die Schüler erkennen, dass sie bei einigen Teilproblemen auf bereits vorgefertigte Standardlösungen in Form von Softwaremustern zurückgreifen können.

## **Inf 11.2.1 Planung und Durchführung kooperativer Arbeitsabläufe**

Die Schüler systematisieren und vertiefen am konkreten Beispiel ihre Kenntnisse über die verschiedenen Schritte bei der Planung und Durchführung eines Softwareprojekts. Zur Koordinierung paralleler Arbeitsgruppen nutzen sie das Semaphorprinzip.

- Projektplanung: Zielsetzung, Arbeitsteilung, Arbeitsgruppen und deren organisatorische Schnittstellen, Ablauf mit Zwischenergebnissen (Meilensteinen)
- Phasen der Softwareentwicklung: Analyse mit Erstellung des Pflichtenhefts, Entwurf, Implementierung, Test, Bewertung und Abnahme
- Problematik der Koordination nebenläufiger Arbeitsabläufe beim Zugriff auf gemeinsame Ressourcen: kritischer Abschnitt, Semaphorprinzip, Verklemmung und Lösungsmöglichkeiten
- Reflexion des Projektverlaufs, Aufwandsabschätzung

## **Inf 11.2.2 Praktische Softwareentwicklung**

Bisher haben die Schüler verschiedene Modellierungstechniken der Informatik einzeln angewandt. Nun erkennen sie, dass eine angemessene Beschreibung größerer Systeme nur durch die kombinierte Verwendung aller bisher erlernten Modellierungstechniken möglich ist. Bei der Implementierung ihrer Modelle setzen sie bekannte Datenstrukturen situationsgerecht ein und achten bei der Gestaltung der Benutzeroberfläche insbesondere auf Benutzungsfreundlichkeit.

- Zusammenspiel der verschiedenen Beschreibungstechniken beim Systementwurf: Datenmodellierung – Ablaufmodellierung – funktionale Modellierung – Objektmodellierung
- Implementierung des Systementwurfs unter Nutzung rekursiver Datenstrukturen; Anwendung des Softwaremusters „Model-View-Controller“
- Test der Komponenten und des Gesamtsystems, Überprüfung der Vollständigkeit und Korrektheit des Systementwurfs
- Dokumentation des Softwareprodukts

# Jahrgangsstufe 12

## Inf 12.1 Formale Sprachen (ca. 16 Std.)

Bisher kennen die Schüler Sprachen vor allem als Mittel zur Kommunikation zwischen Menschen. Ihnen ist bekannt, dass eindeutiges gegenseitiges Verstehen nur dann gewährleistet ist, wenn sich die Kommunikationspartner auf eine gemeinsame Sprache einigen und die zu deren Gebrauch festgelegten Regeln einhalten. Im Rückblick auf das bisherige Arbeiten mit dem Computer wird ihnen deutlich, dass die Verständigung zwischen Mensch und Maschine ebenfalls einen Kommunikationsprozess darstellt, der ähnlichen Regeln unterliegt. Daher betrachten sie zunächst den strukturellen Aufbau einer ihnen bereits bekannten natürlichen Sprache sowie den Aufbau einer künstlichen Sprache. Die Jugendlichen lernen dabei, die Begriffe Syntax und Semantik einer Sprache zu unterscheiden.

Anhand einfacher Beispiele wie Autokennzeichen, E-Mail-Adressen oder Gleitkommazahlen lernen die Schüler den Begriff der formalen Sprache als Menge von Zeichenketten kennen, die nach bestimmten Regeln aufgebaut sind. Zur Beschreibung dieser Regeln verwenden sie Textnotationen oder Syntaxdiagramme und können damit analog zu den natürlichen Sprachen Grammatiken für formale Sprachen definieren. Die Zweckmäßigkeit der streng formalen Beschreibung zeigt sich den Jugendlichen bei der automatischen Überprüfung der syntaktischen Korrektheit von Zeichenketten mithilfe von endlichen Automaten.

Den Schülern wird bewusst, dass nur Vorgänge, die sich mit Mitteln einer formalen Sprache ausdrücken lassen, von einem Computer bearbeitet werden können. Somit stoßen sie über die Theorie der formalen Sprachen auf eine prinzipielle Grenze des Computereinsatzes.

- einfache Beispiele für formale Sprachen über einem Alphabet; Zeichen, Zeichenvorrat (Alphabet), Zeichenkette
- Unterscheidung zwischen Syntax und Semantik, Vergleich zwischen natürlichen und formalen Sprachen
- syntaktischer Aufbau einer formalen Sprache: Grammatik (Terminal, Nicht-terminal, Produktion, Startsymbol)
- Notation formaler Sprachen: Syntaxdiagramm, einfache Textnotation (z. B. Backus-Naur-Form)
- erkennender, endlicher Automat als geeignetes Werkzeug zur Syntaxprüfung für reguläre Sprachen; Implementierung eines erkennenden Automaten

## **Inf 12.2 Kommunikation und Synchronisation von Prozessen (ca. 20 Std.)**

Aus vielen Bereichen der Computernutzung wie beispielsweise dem Homebanking oder einem Buchungssystem für Reisen ist den Schülern die Notwendigkeit zur Zusammenarbeit mehrerer Computer bekannt. Sie sehen ein, dass zur korrekten Verständigung von Computern spezielle formale Regeln (Protokolle) existieren müssen. Anhand der Kommunikation in Rechnernetzen erfahren die Jugendlichen, dass es sinnvoll ist, Kommunikationsvorgänge in verschiedene, aufeinander aufbauende Schichten aufzuteilen.

Die Schüler erkennen z. B. bei der Analyse der Aufgaben eines Mailservers, dass Rechner anfallende Aufträge oft parallel bearbeiten und dennoch einen geordneten Ablauf gewährleisten müssen. Sie erarbeiten Lösungsansätze für die Synchronisation derartiger Vorgänge an Beispielen aus dem täglichen Leben wie der Regelung des Gegenverkehrs durch eine einspurige Engstelle. Beim Übertragen dieser Ansätze auf die Prozesse im Computer wird ihnen bewusst, dass es bei jedem dieser Verfahren bestimmte Sequenzen gibt, die nicht von mehreren Prozessen gleichzeitig ausgeführt werden dürfen.

- Kommunikation zwischen Prozessen, Protokolle zur Beschreibung dieser Kommunikation; Schichtenmodell
- Topologie von Rechnernetzen (Bus, Stern, Ring); Internet als Kombination von Rechnernetzen
- Modellierung einfacher, nebenläufiger Prozesse, z. B. mithilfe eines Sequenzdiagramms; Möglichkeit der Verklemmung
- kritischer Abschnitt; Monitorkonzept zur Lösung des Synchronisationsproblems
- Implementierung eines Beispiels für nebenläufige Prozesse

## **Inf 12.3 Funktionsweise eines Rechners (ca. 17 Std.)**

Am Modell der Registermaschine lernen die Schüler den grundsätzlichen Aufbau eines Computersystems und die Analogie zwischen den bisher von ihnen verwendeten Ablaufmodellen und Maschinenprogrammen kennen. So wird ihnen auch bewusst, dass Möglichkeiten und Grenzen theoretischer algorithmischer Berechnungsverfahren für die reale maschinelle Verarbeitung von Information ebenfalls gelten.

Beispiele zeigen den Schülern, wie einfache Algorithmen auf systemnaher Ebene durch Maschinenbefehle realisiert werden können. Dabei beschränken sich Anzahl und Komplexität der benutzten Maschinenbefehle auf das für die Umsetzung der gewählten Beispiele Wesentliche. Für das Verstehen des Programmablaufs ist insbesondere die in Jahrgangsstufe 10 erlernte Zustandsmodellierung eine große Hilfe. Zur Überprüfung ihrer Überlegungen setzen die Schüler eine Simulationssoftware für die Zentraleinheit ein, die die Vorgänge beim Programmablauf veranschaulicht.

- Aufbau eines Computersystems: Prozessor (Rechenwerk, Steuerwerk), Arbeitsspeicher, Ein- und Ausgabeeinheiten, Hintergrundspeicher; Datenbus, Adressbus und Steuerbus
- Registermaschine als Modell eines Daten verarbeitenden Systems (Datenregister, Befehlsregister, Befehlszähler, Statusregister); Arbeitsspeicher für Programme und Daten (von-Neumann-Architektur), Adressierung der Speicherzellen
- ausgewählte Transport-, Rechen- und Steuerbefehle einer modellhaften Registermaschine; grundsätzlicher Befehlszyklus
- Zustandsübergänge der Registermaschine als Wirkung von Befehlen
- Umsetzung von Wiederholungen und bedingten Anweisungen auf Maschinenebene

## **Inf 12.4 Grenzen der Berechenbarkeit (ca. 10 Std.)**

Nachdem die Jugendlichen bereits erkannt haben, dass sich nicht formalisierbare Aufgabenstellungen der Bearbeitung durch eine Rechenanlage entziehen, bauen sie nun ihre Fähigkeit zur Beurteilung der Einsatzmöglichkeiten maschineller Informationsverarbeitung weiter aus. Sie vertiefen dabei ihr Bewusstsein für grundlegende Einschränkungen, denen jede auch noch so leistungsfähige Rechenanlage unterworfen ist.

Einen Einblick in die praktischen Grenzen der Berechenbarkeit gewinnen die Schüler mithilfe von Aufwandsbetrachtungen an Aufgabenstellungen wie der Wegesuche, die algorithmisch zwar vollständig lösbar sind, bei denen die Ausführung des jeweiligen Algorithmus aber nicht mit vertretbarem Zeitaufwand realisierbar ist. Den Jugendlichen wird deutlich, dass die Sicherheit moderner Verschlüsselungsverfahren auf den praktischen Grenzen der Berechenbarkeit beruht.

Anhand des Halteproblems lernen die Schüler schließlich auch prinzipielle Grenzen der Berechenbarkeit kennen. Sie sehen ein, dass es mathematisch exakt definierbare Probleme gibt, die algorithmisch und damit auch technisch unlösbar sind.

- experimentelle Abschätzung des Laufzeitaufwands typischer Algorithmen und die damit verbundenen Grenzen der praktischen Anwendbarkeit
- hoher Laufzeitaufwand als Schutz vor Entschlüsselung durch systematisches Ausprobieren aller Möglichkeiten (Brute-Force-Verfahren)
- prinzipielle Grenzen der Berechenbarkeit anhand von Plausibilitätsbetrachtungen zum Halteproblem