

Fachlehrplan Gymnasium

Stand: 15.05.2017



SACHSEN-ANHALT

Ministerium für Bildung

Informatik

An der Erarbeitung des Fachlehrplans haben mitgewirkt:

Burgemeister, Maik	Bernburg
Degen, Antje	Hettstedt
Eschrich, Mario	Halle
Grötzsch, Steffi	Halle (Leitung der Fachgruppe)
Dr. Herper, Henry	Otto-von-Guericke-Universität Magdeburg (fachwissenschaftliche Beratung)
Prof. Dr. Müller-Hannemann, Matthias	Martin-Luther-Universität Halle-Wittenberg (fachwissenschaftliche Beratung)

Inhaltsverzeichnis

	Seite
1	Bildung und Erziehung im Fach Informatik 2
2	Entwicklung fachbezogener Kompetenzen 4
2.1	Kompetenzmodell 4
2.2	Inhaltsbereiche 5
2.3	Kompetenzbereiche 12
3	Kompetenzentwicklung in den Schuljahrgängen 14
3.1	Übersicht 14
3.2	Schuljahrgang 9 15
3.3	Schuljahrgang 10 (Einführungsphase) 17
3.4	Schuljahrgänge 11/12 (Qualifikationsphase) 21

1 Bildung und Erziehung im Fach Informatik

In den letzten Jahrzehnten hat die rasante Entwicklung auf dem Gebiet der Informatik wie keine andere Wissenschaft zu gravierenden Veränderungen im gesamtgesellschaftlichen Bereich geführt, deren Auswirkungen auf das tägliche Leben unverkennbar sind. Die Fortschritte in den Teilgebieten der Informatik – der theoretischen, praktischen, angewandten und technischen Informatik – beeinflussen alle Forschungsgebiete. Es entstanden zahlreiche neue interdisziplinäre Wissenschaften wie zum Beispiel Wirtschaftsinformatik, Geoinformatik, Bioinformatik und Medieninformatik. Betrachtet man die Wechselwirkung von Informatik und Gesellschaft so sind auch in Zukunft tiefgreifende Veränderungen in allen Lebensbereichen zu erwarten.

Teilhabe und Teilnahme am gesellschaftlichen Leben

Im Fach Informatik erwerben die Schülerinnen und Schüler informatische Kompetenzen, welche die Grundlage für eine vertiefte berufliche Bildung in allen Berufsfeldern und für die allgemeine Studierfähigkeit darstellen.

Anhand exemplarischer Praxisbeispiele werden vielfältige Bezüge zu Berufen und Studienrichtungen hergestellt.

Die kompetente Nutzung von Informatiksystemen und das Verständnis ihrer Funktionsweise ermöglichen den Schülerinnen und Schülern eine aktive Teilhabe an der gesellschaftlichen Entwicklung auf dem Weg zur global vernetzten Informationsgesellschaft. Sie ist Voraussetzung für die Mitgestaltung der Informationsgesellschaft mit all ihren Chancen und Risiken, Möglichkeiten und Grenzen.

Lebensweltbezogenes Lernen

Die informatische Bildung ermöglicht den Schülerinnen und Schülern in unserer gegenwärtigen und zukünftigen Gesellschaft informatische Sachverhalte zu verstehen und diese sowohl selbstständig als auch im Team zu bewältigen. Sie schließt das grundlegende Verständnis moderner digitaler Hilfsmittel sowie die Beherrschung grundlegender Methoden und Werkzeuge der Informatik, die für die Bewältigung unterschiedlichster Berufs- und Alltagssituationen notwendig sind, ein. Mit der interdisziplinären Anwendung dieser informatischen Bildung als auch der Einbettung fachlicher Zusammenhänge und Erklärungen in lebensweltliche Bezüge der Schülerinnen und Schüler wird ein wesentlicher Beitrag zum Verständnis der Lebenswelt und zur Studien- und Berufsorientierung geleistet.

Das Wahlpflichtfach Informatik trägt zur Ausprägung der Studierfähigkeit und damit zur Allgemeinen Hochschulreife bei, indem die Schülerinnen und Schüler

*Allgemeine
Hochschulreife*

- sich mit dem Konzept, der Implementierung und dem Einsatz von Informatiksystemen für unterschiedliche Anwendungsgebiete beschäftigen,
- eigenständig oder im Team informatische Frage- und Problemstellungen erkennen, analysieren, Lösungswege planen, Lösungen erarbeiten, kontrollieren und auswerten,
- längerfristige Lernprozesse, z. B. bei der Erstellung von Facharbeiten oder der Durchführung von Projekten, praxisnah als auch ergebnisorientiert planen und realisieren sowie
- eigene Problemlösungsprozesse dokumentieren und Arbeitsergebnisse wissenschaftlichen Normen entsprechend darstellen, präsentieren und interpretieren.

Die Informatik ist sowohl Grundlagenwissenschaft als auch angewandte Wissenschaft.

*Wissenschafts-
propädeutisches
Arbeiten*

Das Fach Informatik führt in der Sekundarstufe I und im Besonderen in der Qualifikationsphase in die Wissenschaft Informatik, deren Fachsprache und Denkweisen ein, indem

- die Prinzipien der Informatik exemplarisch an vorhandenen Informatiksystemen betrachtet werden,
- Informations- und Kommunikationstechniken als Medium, Werkzeug und Inhalt des Lernprozesses genutzt werden,
- die Vorgehensweise des Problemlösens unter Verwendung informatischer Methoden erlernt wird und
- praktische Erfahrungen beim Implementieren gefundener Lösungen erworben werden.

2 Entwicklung fachbezogener Kompetenzen

2.1 Kompetenzmodell

Kompetenzmodell Die Schülerinnen und Schüler erwerben im Gymnasium mathematische und naturwissenschaftlich-technische Kompetenzen im Allgemeinen sowie informatische Kompetenzen im Besonderen. Daraus resultieren Ähnlichkeiten in den Kompetenzmodellen. Die benannten Kompetenzen und die damit verbundenen Erklärungen basieren auf den Empfehlungen der Gesellschaft für Informatik (GI) e. V.¹

Inhaltsbereiche:

- Information und Daten
- Algorithmen
- Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Kompetenzbereiche:

- Modellieren und Implementieren
- Begründen und Bewerten
- Strukturieren und Vernetzen
- Kommunizieren und Kooperieren
- Darstellen und Interpretieren

Die Abbildung veranschaulicht die Verflechtung der Inhalts- und Kompetenzbereiche.

¹ GI – Gesellschaft für Informatik e. V. (Hrsg.): Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarschule I. In: LOG IN, 28. Jg. (2008), Heft Nr. 150/151 und GI - Gesellschaft für Informatik e. V. (Hrsg.): Bildungsstandards Informatik für die Sekundarschule II. Beschluss vom 29. Januar 2016.



Abb. 1: Kompetenzmodell für das Fach Informatik

2.2 Inhaltsbereiche

Schülerinnen und Schüler aller Schuljahrgänge

Information und Daten

- verstehen den Zusammenhang von Information und Daten sowie verschiedene Darstellungsformen für Daten,
- verstehen Operationen auf Daten und interpretieren diese in Bezug auf die dargestellte Information,
- führen Operationen auf Daten sachgerecht durch.

Schülerinnen und Schüler aller Schuljahrgänge

Algorithmen

- kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten, lesen und interpretieren gegebene Algorithmen,
- entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar.

Schülerinnen und Schüler aller Schuljahrgänge

Sprachen und Automaten

- nutzen formale Sprachen zur Interaktion mit Informatiksystemen und zum Problemlösen,
- analysieren und modellieren Automaten.

- Informatiksysteme* Schülerinnen und Schüler aller Schuljahrgänge
- verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise,
 - wenden Informatiksysteme zielgerichtet an,
 - erschließen sich weitere Informatiksysteme.

- Informatik, Mensch und Gesellschaft* Schülerinnen und Schüler aller Schuljahrgänge
- benennen Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung,
 - nehmen Entscheidungsfreiheiten im Umgang mit Informatiksystemen wahr und handeln in Übereinstimmung mit gesellschaftlichen Normen,
 - reagieren angemessen auf Risiken bei der Nutzung von Informatiksystemen.

Die Fachwissenschaft Informatik mit ihren Anwendungen nimmt eine zentrale Stellung in der gesellschaftlichen Entwicklung ein.

Im Wahlpflichtfach Informatik werden in den Schuljahrgängen 9 bis 12 Kompetenzen auf grundlegendem Anforderungsniveau angestrebt. In einigen Inhaltsbereichen werden darüber hinaus erweiterte Kompetenzen erworben, die sich aus aktuellen und zu erwartenden Entwicklungen ergeben.

Dem zehnten Schuljahrgang kommt in seiner Gelenkfunktion mit dem Abschluss der Sekundarstufe I und der Einführungsphase zur Vorbereitung auf die Qualifikationsphase eine besondere Rolle zu. Diese spiegelt sich unter anderem im Erwerb der Kompetenzen im Inhaltsbereich Information und Daten und den entsprechenden Schwerpunkten der Schuljahrgänge 10 und 11 wider. Ausgehend von einer ikonischen Programmiersprache werden über die textbasierte Implementierung im Schuljahrgang 10 die Fähigkeiten und Fertigkeiten bis zur objektorientierten Softwareentwicklung weiterentwickelt.

Im Schuljahrgang 12 werden im Rahmen eines Softwareprojektes erworbene Kompetenzen zusammengeführt und vertieft.

Am Ende der	
Einführungsphase	Qualifikationsphase
verstehen die Schülerinnen und Schüler den Zusammenhang von Information und Daten sowie verschiedene Darstellungsformen für Daten. Die Schülerinnen und Schüler können:	
<ul style="list-style-type: none"> – Informationen in unterschiedlicher Form darstellen, – Daten im Kontext der repräsentierten Information interpretieren, – die Datentypen Text, Zahl und Wahrheitswert verwenden, – Strukturierungsmöglichkeiten von Daten zum Zusammenfassen gleichartiger und unterschiedlicher Elemente zu einer Einheit verwenden. 	<ul style="list-style-type: none"> – Daten hinsichtlich ihrer Struktur analysieren, – Information als Daten mit Datentypen und in Datenstrukturen abbilden.
verstehen die Schülerinnen und Schüler Operationen auf Daten, interpretieren diese in Bezug auf die dargestellte Information und können:	
<ul style="list-style-type: none"> – arithmetische und logische Operationen verwenden, – grundlegende Operationen zum Zugriff auf die Bestandteile strukturierter Daten verwenden. 	<ul style="list-style-type: none"> – zwischen Zeichen, Daten und Information, sowie zwischen Syntax und Semantik unterscheiden.
führen die Schülerinnen und Schüler Operationen auf Daten sachgerecht durch und können:	
<ul style="list-style-type: none"> – Datentypen und Operationen formal darstellen und sie sachgerecht nutzen. 	<ul style="list-style-type: none"> – Operationen auf statischen und dynamischen Datenstrukturen verwenden, modellieren und implementieren, – zu einem Realitätsausschnitt ein Datenmodell erstellen, – eine Abfragesprache zur Anzeige und Manipulation von Daten verwenden und die Daten interpretieren, – Operationen auf komplexen Datenstrukturen verwenden, modellieren und implementieren, – zu einem Ausschnitt der Lebenswelt mit komplexen Beziehungen eine Datenbank entwickeln.

Algorithmen

<p>Am Ende der</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; text-align: center; border: none;">Einführungsphase</td> <td style="width: 50%; text-align: center; border: none;">Qualifikationsphase</td> </tr> </table>		Einführungsphase	Qualifikationsphase
Einführungsphase	Qualifikationsphase		
<p>kennen die Schülerinnen und Schüler Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten, lesen gegebene Algorithmen und interpretieren diese. Die Schülerinnen und Schüler können:</p>			
<ul style="list-style-type: none"> – Handlungsvorschriften aus dem Alltag und für das Arbeiten mit Informatiksystemen benennen, formulieren, lesen und interpretieren, – die wesentlichen Eigenschaften von Algorithmen überprüfen, – formale Darstellungen von Algorithmen lesen und sie in Programme umsetzen. 	<ul style="list-style-type: none"> – gegebene Programme hinsichtlich der Grundkonzepte, einschließlich Variable, Referenz, Schachtelung und funktionale Zerlegung, analysieren. 		
<p>entwerfen und realisieren die Schülerinnen und Schüler Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar. Die Schülerinnen und Schüler können:</p>			
<ul style="list-style-type: none"> – die algorithmischen Grundbausteine formal darstellen, – Variablen und Wertzuweisungen verwenden, – Algorithmen entwerfen, implementieren, testen und beurteilen, – Quelltexte von Programmen nach Vorgaben modifizieren und ergänzen. 	<ul style="list-style-type: none"> – Algorithmen entwerfen und sie in geeigneter Form darstellen, – algorithmische Grundbausteine in Softwareprojekten verwenden, – Softwarebibliotheken oder bereitgestellte Module bei der Implementierung von Algorithmen verwenden, – Programme systematisch testen und überarbeiten. 		

Am Ende der	
Einführungsphase	Qualifikationsphase
nutzen die Schülerinnen und Schüler formale Sprachen zur Interaktion mit Informatiksystemen und zum Problemlösen. Die Schülerinnen und Schüler können:	
<ul style="list-style-type: none"> – Problemlösungen in einer Programmiersprache angeben, – die Begriffe Syntax und Semantik unterscheiden und sie an Beispielen erläutern, – Objekte der jeweiligen Anwendung in einer geeigneten Form darstellen, – Fehlermeldungen bei der Arbeit mit Informatiksystemen interpretieren und sie zur Fehlerbehebung nutzen. 	<ul style="list-style-type: none"> – formale mit natürlichen Sprachen vergleichen, – Sprachdefinitionen (z. B. Grammatiken, Syntaxdiagramme) zur Analyse formaler Sprachen verwenden.
analysieren und modellieren die Schülerinnen und Schüler Automaten und können:	
<ul style="list-style-type: none"> – das Prinzip der Eingabe, Verarbeitung und Ausgabe von Daten (EVA-Prinzip) als grundlegendes Arbeitsprinzip von Informatiksystemen erläutern, – Eingaben und Ausgaben realer Automaten unterscheiden, – unterschiedliche Zustände realer Automaten identifizieren, – Zustandsübergänge realer Automaten und die Eingaben, die sie ausgelöst haben, beschreiben, – Automaten analysieren und sie zustandsorientiert modellieren – einfache Zustandsdiagramme interpretieren, – den Zusammenhang zwischen Automaten und Sprachen erläutern. 	<ul style="list-style-type: none"> – Prozesse modellieren und implementieren.

Am Ende der	
Einführungsphase	Qualifikationsphase
verstehen die Schülerinnen und Schüler die Grundlagen des Aufbaus von Informatiksystemen sowie deren Funktionsweise und können:	
<ul style="list-style-type: none"> – wesentliche Komponenten von Informatiksystemen benennen und charakterisieren, – Bestandteile eines Informatiksystems der Eingabe, der Verarbeitung und der Ausgabe zuordnen, – Daten speichern und Arten der Speicher unterscheiden, – Betriebssystem und Anwendersoftware unterscheiden, – lokale von globalen Netzen unterscheiden. 	<ul style="list-style-type: none"> – wesentliche Komponenten der Architektur gegebener Informatiksysteme und die damit verbundenen Prozesse beschreiben und erklären.
wenden die Schülerinnen und Schüler Informatiksysteme zielgerichtet an und können:	
<ul style="list-style-type: none"> – Dienste des Betriebssystems zweckgerichtet benutzen, – Dateiformate unterscheiden, – in Netzwerken und mit Internetdiensten arbeiten, – ein Netzwerk mithilfe geeigneter Strukturierungs- und Darstellungsmethoden entwickeln. 	<ul style="list-style-type: none"> – die Kommunikation und die Datenhaltung in vernetzten Systemen analysieren und diese auch unter den Gesichtspunkten des Datenschutzes und der Datensicherheit beurteilen.
erschließen sich die Schülerinnen und Schüler weitere Informatiksysteme und können:	
<ul style="list-style-type: none"> – Informatiksysteme in Alltagsgeräten erkennen. 	<ul style="list-style-type: none"> – komplexe Informatiksysteme in ihrer Lebenswelt erkennen und analysieren.
modellieren und implementieren die Schülerinnen und Schüler Software als Bestandteil von Informatiksystemen und können:	
<ul style="list-style-type: none"> – Klassen und Objekte, Attribute und Methoden darstellen, – Eigenschaften von Objekten über Attribute und Methoden verändern. 	<ul style="list-style-type: none"> – den objektorientierten Ansatz verwenden, indem sie Klassen mit ihren Attributen, Methoden und Beziehungen modellieren und implementieren, – Konzepte und Methoden der Softwareentwicklung zur Gestaltung und Entwicklung von Informatiksystemen auch unter Berücksichtigung von Aspekten der Softwareergonomie anwenden, – Informatiksysteme auf Basis von Qualitätskriterien gestalten.

Am Ende der	
Einführungsphase	Qualifikationsphase
benennen die Schülerinnen und Schüler Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung. Die Schülerinnen und Schüler können:	
<ul style="list-style-type: none"> – die historische Entwicklung der Informatik im gesellschaftlichen Kontext erkennen. 	<ul style="list-style-type: none"> – die Veränderung der Lebens- und Arbeitswelt durch Informatiksysteme beurteilen, – Wechselwirkungen zwischen Informatiksystemen, Individuen und Gesellschaft analysieren und beschreiben.
nehmen die Schülerinnen und Schüler Entscheidungsfreiheiten im Umgang mit Informatiksystemen wahr, handeln in Übereinstimmung mit gesellschaftlichen Normen und können:	
<ul style="list-style-type: none"> – Umgangsformen bei elektronischer Kommunikation beachten und auf die Persönlichkeitsrechte anderer achten, – die Notwendigkeit einer verantwortungsvollen Nutzung von Informatiksystemen erkennen. 	<ul style="list-style-type: none"> – Chancen, Risiken und Missbrauchsmöglichkeiten von Informatiksystemen beschreiben, – die gesellschaftlichen Folgen der Einführung und Nutzung von Informatiksystemen beurteilen und bewerten.
reagieren die Schülerinnen und Schüler angemessen auf Risiken bei der Nutzung von Informatiksystemen und können:	
<ul style="list-style-type: none"> – die leichte Manipulierbarkeit digitaler Daten bewusst erfassen, – die Unsicherheit einfacher Verschlüsselungsverfahren erkennen. 	<ul style="list-style-type: none"> – Verfahren zur Sicherung von Vertraulichkeit und Integrität von Daten verwenden und beschreiben, – Rückschlüsse auf das eigene Verhalten beim Einsatz von Informatiksystemen ziehen.

2.3 Kompetenzbereiche

<i>Modellieren und Implementieren</i>	Schülerinnen und Schüler aller Schuljahrgänge <ul style="list-style-type: none">– erstellen informatische Modelle zu gegebenen Sachverhalten,– implementieren Modelle mit geeigneten Werkzeugen,– reflektieren Modelle und ihre Implementation.
<i>Begründen und Bewerten</i>	Schülerinnen und Schüler aller Schuljahrgänge <ul style="list-style-type: none">– stellen Fragen und äußern Vermutungen über informatische Sachverhalte,– begründen Entscheidungen bei der Nutzung von Informatiksystemen,– wenden Kriterien zur Bewertung informatischer Sachverhalte an.
<i>Strukturieren und Vernetzen</i>	Schülerinnen und Schüler aller Schuljahrgänge <ul style="list-style-type: none">– strukturieren Sachverhalte durch zweckdienliches Zerlegen und Anordnen,– erkennen und nutzen Verbindungen innerhalb und außerhalb der Informatik.
<i>Kommunizieren und Kooperieren</i>	Schülerinnen und Schüler aller Schuljahrgänge <ul style="list-style-type: none">– kommunizieren fachgerecht über informatische Sachverhalte,– kooperieren bei der Lösung informatischer Probleme,– nutzen geeignete Werkzeuge zur Kommunikation und Kooperation.
<i>Darstellen und Interpretieren</i>	Schülerinnen und Schüler aller Schuljahrgänge <ul style="list-style-type: none">– interpretieren unterschiedliche Darstellungen von Sachverhalten,– veranschaulichen informatische Sachverhalte,– wählen geeignete Darstellungsformen aus.

Im Wahlpflichtfach Informatik werden über die fachspezifischen Kompetenzen hinaus übergreifende Schlüsselkompetenzen entwickelt, die für eine aktive Teilnahme und Teilhabe am gesellschaftlichen Leben notwendig sind.

*Beitrag zur
Entwicklung der
Schlüssel-
kompetenzen*

Die Auseinandersetzung mit den Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung stärkt das Verständnis für wirtschaftliche, soziale und gesellschaftliche Zusammenhänge. Der Umgang mit Informatiksystemen erfolgt in Übereinstimmung mit gesellschaftlichen Normen und berücksichtigt die Persönlichkeitsrechte anderer. Damit wird ein Beitrag zur Entwicklung eines Rechtsbewusstseins und zur Einhaltung gesellschaftlicher Normen geleistet. Schülerinnen und Schüler erwerben umfassende Kompetenzen im logischen Denken und stärken ihre Abstraktionsfähigkeit. Sie sind in der Lage, Lösungsstrategien und komplexe Abläufe formal exakt darzustellen und zu interpretieren.

Die Arbeit in den verschiedensten Projekten in allen Schuljahrgängen erfordert den norm-, sach-, situations- und adressatengerecht Einsatz von Sprache in mündlicher und schriftlicher Form und trägt somit zur Entwicklung der Kommunikationskompetenz bei. Schülerinnen und Schüler entwickeln für sich im Rahmen von Projekten ein effizientes Zeitmanagement. Das schließt Selbstorganisation, Reflexion des Lehrens und Lernens sowie das Ableiten von Rückschlüssen für sich selbst und für die Lerngruppe bezüglich der Ziele, Strategien, Ergebnisse und Lernhaltungen im Projektverlauf ein. Die Bereitschaft, Hindernisse im Verlauf des Problemlösungszyklus in Kooperation mit anderen zu überwinden, stärkt die Sozialkompetenz. Das Bearbeiten komplexer Aufgabenstellungen erfordert und fördert kreative Denkansätze.

Die Schülerinnen und Schüler gewinnen, verarbeiten, bewerten und präsentieren Informationen, tauschen diese aus und verwenden sie für eigenständiges Lernen grundsätzlich unter Nutzung digitaler Werkzeuge und Endgeräte. Gleichzeitig wird das Verständnis für die den digitalen Werkzeugen zugrunde liegenden Funktionsweisen erworben. Diese Kompetenzen sind die Basis für mögliche Entwicklungen neuer Werkzeuge.

*Kompetenzen im
Umgang mit
digitalen
Werkzeugen und
Endgeräten*

3 Kompetenzentwicklung in den Schuljahren

3.1 Übersicht

Schuljahrgänge	Kompetenzschwerpunkte
Schuljahrgang 9	<ul style="list-style-type: none">– Algorithmen interpretieren und entwickeln– Grundlagen von Informatiksystemen verstehen
Schuljahrgang 10 (Einführungsphase)	<ul style="list-style-type: none">– Algorithmen- und Datenstrukturen implementieren und testen– Informatiksysteme in der Lebens- und Arbeitswelt verstehen
Schuljahrgänge 11/12 (Qualifikationsphase)	<ul style="list-style-type: none">– Kurs 1: Objektorientiertes Modellieren– Kurs 2: Daten erfassen, strukturieren und verarbeiten– Kurs 3: Software Engineering und Projektarbeit– Kurs 4: Aktuelle Entwicklungen der Informatik im gesellschaftlichen Kontext

3.2 Schuljahrgang 9

Kompetenzschwerpunkt: Algorithmen interpretieren und entwickeln	
Modellieren und Implementieren	<ul style="list-style-type: none"> – Handlungsabläufe formulieren – Algorithmen entwerfen und verbal darstellen – Variable als Speicherplatz beschreiben und anwenden – Algorithmen in eine ikonische Programmiersprache implementieren – reale endliche Automaten mit Ausgabe analysieren und das zugehörige Automatenmodell entwerfen
Begründen und Bewerten	<ul style="list-style-type: none"> – Algorithmen testen und weiterentwickeln – gegebene Automaten bezüglich ihrer Praxisrelevanz bewerten – Entscheidungen im Rahmen eines Problemlösungszyklus begründen
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – Problemstellungen analysieren, in Teilprobleme zerlegen und Lösungsansätze strukturieren – Algorithmen strukturieren – Auswirkungen von Algorithmen auf die Lebenswelt exemplarisch erkennen
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – einen Problemlösungszyklus reflektieren – Algorithmen diskutieren, dokumentieren und präsentieren – Auswirkungen von Algorithmen auf ihre eigene Lebenswelt reflektieren – reale Automaten beschreiben
Darstellen und Interpretieren	<ul style="list-style-type: none"> – Algorithmen in der Lebenswelt in unterschiedlichen Darstellungsformen erkennen und interpretieren – algorithmische Bausteine benennen, darstellen und auf Handlungsabläufe anwenden – Automatentafel und Zustandsdiagramm endlicher Automaten mit Ausgabe interpretieren und wechselseitig übertragen
Grundlegende Wissensbestände	
<ul style="list-style-type: none"> – Begriffe: Algorithmus, Programmiersprache – Eigenschaften von Algorithmen – algorithmische Grundbausteine (Anweisung, Anweisungsfolge, Verzweigung, Wiederholung) – Problemlösungszyklus – Modell endlicher Automat mit Ausgabe – Automatentafel und Zustandsdiagramm: zwei Darstellungsformen zur Beschreibung der Zustandsüberföhrungsfunktion und der Ausgabefunktion 	
Mögliche Projekte	
<ul style="list-style-type: none"> – Algorithmen erleben – implementieren von algorithmischen Problemlösungen – kommunizieren mit externen Geräten (Einbinden von Sensoren und Aktoren) – implementieren von endlichen Automaten mit Ausgabe 	

Kompetenzschwerpunkt: Grundlagen von Informatiksystemen verstehen	
Modellieren und Implementieren	<ul style="list-style-type: none"> – Informationen auf Daten abbilden – reale Rechner analysieren und ein Funktionsmodell erstellen
Begründen und Bewerten	<ul style="list-style-type: none"> – Dateiformate benennen sowie analysieren und diesen Anwendungen zuordnen – Lebensdauer von Daten beurteilen – technische Parameter von Hardwarekomponenten bewerten – Informatiksysteme im historischen Kontext einordnen
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – Zusammenhänge zwischen Daten und Information darstellen – wesentliche Hardware-, Software- und Netzwerkkomponenten benennen sowie deren Aufgaben erläutern
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – Informatiksysteme in ihrer Lebenswelt erkennen, benennen und untersuchen – Daten lokal und in Netzwerken verwalten – Aufgaben in Netzwerken bearbeiten und grundlegende Verhaltensnormen beachten – die Funktionsweise einfacher Grundschaltungen beschreiben und erklären
Darstellen und Interpretieren	<ul style="list-style-type: none"> – Einheiten für die Datenmenge auf die Speicherkapazität von Datenträgern, Dateigrößen und Übertragungsvolumen anwenden – Betriebssysteme vergleichen und Grundfunktionen erläutern
Grundlegende Wissensbestände	
<ul style="list-style-type: none"> – Meilensteine der Informatik – Maßeinheiten und Kodierung von Daten – Zahlensysteme der Informatik – Informationsbegriff (Information als kontextbezogene Interpretation von Daten) – Dateiformat als spezifische Beschreibung und Darstellung von Daten innerhalb von Dateien – Datensicherheit als Beschreibung aller Maßnahmen, um Daten jeglicher Art gegen Verlust, Manipulationen und andere Bedrohungen zu sichern – verschiedene Typen von Datenspeichern – logische Funktionen und Grundbausteine – Grundschaltungen (z. B. Flipflop, Halb- und Volladdierer) – EVA-Prinzip und zugehörige Hardwarekomponenten – Rechnermodell nach J. v. Neumann – Dateiverwaltung – Prozesse und Prozessverwaltung – Intranet und Internet 	
Mögliche Projekte	
<ul style="list-style-type: none"> – Simulation und Implementation von Grundschaltungen – Analyse von Dateiformaten – Untersuchen eines realen Informatiksystems 	

3.3 Schuljahrgang 10 (Einführungsphase)

Kompetenzschwerpunkt: Algorithmen- und Datenstrukturen implementieren und testen	
Modellieren und Implementieren	<ul style="list-style-type: none"> – Methoden einer Klasse als Möglichkeit der Abfrage und Änderung von Attributen erfassen – Daten mittels geeigneter Datentypen abbilden und zugehörige Operationen anwenden – zwischen einfachen und strukturierten Datentypen unterscheiden und diese anwenden – das Variablenkonzept anwenden – algorithmische Grundbausteine in eine textbasierte Programmiersprache implementieren – grafisch dargestellte Algorithmen testen und implementieren – Sortier- und Verschlüsselungsverfahren implementieren und testen
Begründen und Bewerten	<ul style="list-style-type: none"> – gewählte Klassenstrukturen begründen – die Notwendigkeit von Sortier- und Verschlüsselungsverfahren begründen – Testergebnisse werten und daraus Veränderungen ableiten – Entscheidungen im Rahmen eines Problemlösungszyklus begründen und in Teilen bewerten
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – Klassen als Möglichkeit der Datenstrukturierung in Form von Eigenschaften nutzen – Objekte als Instanzen einer Klasse mit konkreten Daten beschreiben – Problemlösungsstrategien strukturieren und beschreiben
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – Algorithmen sachgerecht dokumentieren und präsentieren – einfache Formen der Fehlerbehandlung in Programmen diskutieren und nutzen
Darstellen und Interpretieren	<ul style="list-style-type: none"> – Klasse, Objekt, Attribut und Methode mittels der Modellierungssprache UML darstellen – Algorithmen unter Verwendung algorithmischer Grundbausteine grafisch darstellen – Fehlermitteilungen interpretieren
Grundlegende Wissensbestände	
<ul style="list-style-type: none"> – Begriffe: Klasse, Objekt, Instanz, Attribut, Methode – UML-Diagramme: Klassendiagramm, Objektdiagramm – Syntax einer Programmiersprache und formale Beschreibungsformen – Algorithmische Grundbausteine sowie deren formale Beschreibungsformen und Entsprechungen in einer Programmiersprache – einfache und strukturierte Datentypen – Variablen und Konstanten – Zuweisungs- und Vergleichsoperatoren, arithmetische und logische Operatoren – ein elementares Sortierverfahren und ein elementares Verschlüsselungsverfahren 	

Mögliche Projekte

Umsetzung der Phasen eines Problemlösungszyklus am Beispiel:

- eines weiteren Sortierverfahrens oder
- eines weiteren Verschlüsselungsverfahrens oder
- eines Suchverfahrens

Kompetenzschwerpunkt: Informatiksysteme in der Lebens- und Arbeitswelt verstehen	
Modellieren und Implementieren	<ul style="list-style-type: none"> – endliche Automaten ohne Ausgabe analysieren und das zugehörige Automatenmodell entwerfen – Regeln für einfache formale Sprachen aufstellen und diese als Akzeptor implementieren – Worte einer formalen Sprache aus der Kenntnis ihrer Grammatik erzeugen – Heim- bzw. Schulnetzwerke exemplarisch modellieren und simulieren – ausgewählte Netzwerkdienste unter Nutzung zugehöriger Protokolle in einer geeigneten Umgebung einrichten und nutzen
Begründen und Bewerten	<ul style="list-style-type: none"> – reale Automaten aus ihrer Lebenswelt diskutieren und bewerten – ein Wort aus Terminalsymbolen eines Alphabetes zu einer vorgegebenen Sprache als zugehörig erkennen oder verneinen – die Existenz von Fehlerzuständen in Automaten begründen – die Notwendigkeit einer auf exakten Regeln basierenden Sprache, welche eindeutig maschinell ausführbar ist, begründen – exemplarisch Grenzen automatisierter Systeme erkennen – die Notwendigkeit von Schutz- und Sicherheitsmaßnahmen in Netzwerken begründen und zielgerichtet anwenden – Verhaltensweisen in Netzwerken reflektieren und Schlussfolgerungen für das eigene Handeln ableiten
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – gegebene Informatiksysteme analysieren sowie Daten- und Funktionsstrukturen ableiten – zwischen natürlichen und formalen Sprachen unterscheiden – die Funktionsweise von TCP/IP-Netzwerken am Beispiel des Schulnetzwerkes erläutern, strukturieren und simulieren – Dienste, zugehörige Protokolle und Ports exemplarisch erläutern
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – Eigenschaften endlicher Automaten erläutern – endliche Automaten mit und ohne Ausgabe vergleichen und weitere Anwendungsfälle diskutieren – Anwendungsfälle formaler Sprachen nennen – Bedeutung der Vernetzung von digitalen Systemen kennen sowie Anwendungsfälle nennen und erläutern – Funktionalität von Diensten unter Nutzung eines Schichtenmodells exemplarisch beschreiben – lokale und netzwerkbasierte Datenverwaltung vergleichen und anwenden
Darstellen und Interpretieren	<ul style="list-style-type: none"> – die Funktionalität von Informatiksystemen mittels geeigneter Diagramme abbilden – die Überföhrungsfunktion endlicher Automaten ohne Ausgabe in verschiedenen Formen darstellen, interpretieren und wechselseitig übertragen – die formale Sprache, die ein endlicher Automat akzeptiert, in geeigneter Form darstellen – einfache formale Sprachen exemplarisch mittels Syntaxdiagrammen darstellen

Grundlegende Wissensbestände

- Modelle endlicher Automaten
- Grenzen automatisierter Systeme (z. B. Turing Test)
- Begriffe: Terminalsymbol, Nichtterminalsymbol, Alphabet, Wort, Grammatik, Semantik, Syntax, Syntaxdiagramme
- TCP/IP–Netzwerke (Aufbau, Funktionsweise, Schichtenmodell)
- aktive und passive Netzwerkkomponenten
- Adressierung, Adressauflösung und Routing in Netzwerken
- Kommunikation in kabelgebundenen und drahtlosen Netzwerken
- Peer-to-Peer-Netzwerk, Client-Server-Netzwerk
- Schutz und Sicherheit in Netzwerken (Firewall, Virenschutz, Backup)
- grundlegende Verhaltensregeln in Netzwerken

Mögliche Projekte

- Netzwerkstrukturen mit geeigneter Software simulieren oder real aufbauen
- analysieren eines Smart-Home

3.4 Schuljahrgänge 11/12 (Qualifikationsphase)

Kurs 1: Objektorientiertes Modellieren	
Modellieren und Implementieren	<ul style="list-style-type: none"> – Problemstellungen unter Verwendung von Anwendungsfalldiagrammen analysieren und dokumentieren – objektorientierte Analysen durchführen sowie beteiligte Objekte, ihre Eigenschaften, Operationen und vorhandene Beziehungen erkennen und Klassen ableiten – Prozesse mithilfe von Aktivitäts- oder Zustandsdiagrammen modellieren – Klassen mit ihren Methoden in eine Programmiersprache implementieren – Objekte erzeugen, in dynamischen Datenstrukturen verwalten, auf externen Datenträgern speichern und von diesen laden – die Kommunikation zwischen und innerhalb von Objekten organisieren – Module und Bibliotheken in eigene Problemlösungen integrieren
Begründen und Bewerten	<ul style="list-style-type: none"> – die gewählte Abstraktion begründen – eigene Lösungen bewerten und weitere Arbeitsschritte ableiten – gegebene Programme hinsichtlich ihrer Grundkonzepte analysieren
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – einen objektorientierten Entwurf (Design) in der Modellierungssprache UML erstellen – das Geheimnisprinzip und das Vererbungskonzept von Klassen auf einfache Sachverhalte anwenden – Probleme in Teilprobleme exemplarisch zerlegen
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – Schnittstellen entsprechend vereinbarter Regeln definieren – UML–Diagramme als Kommunikationsmittel in der Gruppe nutzen
Darstellen und Interpretieren	<ul style="list-style-type: none"> – Klassen und deren Beziehungen in UML darstellen – Klassendefinitionen und Klassenbeschreibungen interpretieren – Fehlermitteilungen interpretieren und unter Verwendung geeigneter Hilfesysteme Lösungsstrategien entwerfen
Grundlegende Wissensbestände	
<ul style="list-style-type: none"> – UML-Diagramme (Aktivitäts- oder Zustandsdiagramm) – Parameterliste (Wert- und Referenzparameter) – Erzeugen und Entfernen von Objekten (Konstruktor: Aufbau, Funktion, Überladung und ggf. Destruktor) – Gültigkeitsbereiche, Zugriffsrechte und Integrität – Überladen von Operationen – Assoziationen, Vererbung – Module und Bibliotheken – Gestaltung grafischer Benutzungsoberflächen – Testverfahren und deren Anwendung 	
Mögliche Projekte	
<p>Unterrichtsbegleitend sind u. a. folgende Projekte denkbar:</p> <ul style="list-style-type: none"> – Aufbau einer Simulation (z. B. Ampelschaltung, endlicher Automat, Warteschlangensysteme) – Entwicklung eines Computerspiels – Entwicklung einer verteilten Anwendung 	

Kurs 2: Daten erfassen, strukturieren und verarbeiten	
Modellieren und Implementieren	<ul style="list-style-type: none"> – für anwendungsrelevante Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten ermitteln – relationale Datenbankschemata unter Nutzung des Entity-Relationship-Modells entwerfen – ein Entity-Relationship-Modell modifizieren – Primär- und Sekundärschlüssel bestimmen – relationale Datenbankschemata bis zur dritten Normalform überführen – eine Abfragesprache anwenden
Begründen und Bewerten	<ul style="list-style-type: none"> – die Verwendung von Datenstrukturen begründen – Datenbankmodelle bewerten
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – Daten in geeigneter Form strukturieren und verwalten – Relationen zwischen Entitäten erkennen und diese in geeigneter Form darstellen
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – Entity-Relationship-Diagramme diskutieren und erstellen – eine Datenbankmodellierung analysieren und erläutern – die Eigenschaften eines normalisierten Datenbankschemas erläutern
Darstellen und Interpretieren	<ul style="list-style-type: none"> – Daten in verschiedenen Formen darstellen und bestehende Formen interpretieren – Entitäten, ihre Attribute und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch darstellen
Grundlegende Wissensbestände	
<ul style="list-style-type: none"> – Zusammenhang von Informationen und Daten – Datenbanksystem (Datenbank, Datenbankverwaltungssystem, Frontend, Backend) – Relationale Datenbank – Entity-Relationship-Diagramm – Entität, Attribute, Relation und Kardinalität – Normalisierung einer Datenbank – Schnittstellen von Datenbanken – Abfragesprache zur Anzeige und Manipulation von Daten 	
Mögliche Projekte	
<ul style="list-style-type: none"> – alltagsrelevante Datenbanken modellieren – Vergleich unterschiedlicher Strukturierungsansätze 	

Kurs 3: Software Engineering und Projektarbeit	
Modellieren und Implementieren	<ul style="list-style-type: none"> – selbstständig Analysen im Rahmen eines Problemlösungszyklus durchführen und daraus benötigte Objekte und Klassen ableiten – die Implementierung der Teillösungen selbstständig oder mithilfe der Teampartner realisieren – Teillösungen zu einer Gesamtlösung zusammenführen – Teillösungen sowie Endprodukt verifizieren und validieren
Begründen und Bewerten	<ul style="list-style-type: none"> – Entscheidungen im Rahmen eines Problemlösungszyklus begründen und bewerten – dem Auftraggeber das gewählte Design und die Spezifikationen auf Basis von Qualitätskriterien erläutern und begründen – notwendige Abstraktionen selbstständig vornehmen und begründen – die Auswahl der genutzten Algorithmen begründen und hinsichtlich der Zielführung und Effizienz bewerten – die Datenhaltung in vernetzten Systemen analysieren und diese unter den Gesichtspunkten des Datenschutzes und der Datensicherheit beurteilen
Strukturieren und Vernetzen	<ul style="list-style-type: none"> – ein Problem in Teilprobleme zerlegen und notwendige Schnittstellen vereinbaren – die Teilprobleme so aufteilen, dass Synergieeffekte nutzbar werden – Aspekte der Softwareergonomie umsetzen
Kommunizieren und Kooperieren	<ul style="list-style-type: none"> – Projekte organisieren und bearbeiten – geeignete Quellen zur Vorbereitung und Durchführung der Projektarbeit selbstständig auswählen und nutzen – Abstimmung mit dem Auftraggeber (z. B. Lehrkraft) durchführen und daraus Zielsetzungen für das Endprodukt ableiten – Regeln zur Lösung von Teilproblemen, zur Festlegung von Klassen und Schnittstellen sowie zum Zeitmanagement diskutieren und vereinbaren – projektbegleitend Zwischenergebnisse dokumentieren, kommunizieren und daraus Schlussfolgerungen für das weitere Vorgehen ableiten
Darstellen und Interpretieren	<ul style="list-style-type: none"> – Abläufe des Software-Engineerings mittels verschiedener Diagramme veranschaulichen und interpretieren – eine begleitende Dokumentation des gesamten Projektes in textlicher und grafischer Form (UML-Notation) erstellen – Präsentation auf der Grundlage der Dokumentation des Projektes erstellen und die Projektarbeit verteidigen
Grundlegende Wissensbestände	
<ul style="list-style-type: none"> – Methoden des Software-Engineerings – Dokumentation von Projekten – Projektplanung und -organisation – Spezifikation der Anforderungen – technischer Entwurf der Software – Implementierung und Test – Qualitätssicherung 	
Mögliche Projekte	
<ul style="list-style-type: none"> – Projekte aus dem Erfahrungsfeld der Schülerinnen und Schüler 	

Kurs 4: Aktuelle Entwicklungen der Informatik im gesellschaftlichen Kontext	
Modellieren und Implementieren	– wesentliche Aspekte von komplexen Informatikinhalten erkennen und modellieren
Begründen und Bewerten	– Chancen, Risiken und Missbrauchsmöglichkeiten von Informatiksystemen beschreiben und bewerten – Möglichkeiten und Grenzen in einer von Informatiksystemen dominierten Welt erkennen und bewerten
Strukturieren und Vernetzen	– gegebene Informatiksysteme analysieren und diskutieren – Rückschlüsse auf das eigene Verhalten beim Einsatz von Informatiksystemen ziehen
Kommunizieren und Kooperieren	– gesellschaftliche Folgen der Einführung und Nutzung von Informatiksystemen diskutieren und Handlungsstrategien ableiten
Darstellen und Interpretieren	– Wechselwirkungen zwischen Informatiksystemen, Individuen und Gesellschaft darstellen und beschreiben
Grundlegende Wissensbestände	
<p>Bearbeitet werden ein oder mehrere Themenfelder zu aktuellen Entwicklungen in der Informatik. Beispielhaft seien folgende genannt:</p> <ul style="list-style-type: none"> – aktuelle Informatiksysteme und Technikfolgenabschätzung – Gestaltung von Informatiksystemen – virtuelle Welten (z. B. 3D-Modellierung) 	
Mögliche Projekte	
<ul style="list-style-type: none"> – Aufbau eines Smart-Home – Anwendung von BigData – Simulationen 	