



Deutsche Schule Lissabon
Schulcurriculum

Informatik

– Jahrgangsstufen 10-12 –

Stand: März 2018

Inhaltsverzeichnis

1	Vorwort.....	3
1.1	Rahmenbedingungen an der DSL	3
2	Aufgaben und Ziele des Fachs Informatik.....	4
2.1	Fächer des mathematisch-naturwissenschaftlich-technischen Aufgabenfeld.....	4
2.2	Das Fach Informatik.....	4
2.3	Methodik des Informatik-Unterricht	5
3	Kompetenzbereiche, -erwartungen und Inhaltsfelder	6
3.1	Kompetenzbereiche	6
3.2	Inhaltsfelder	7
3.3	Kompetenzerwartungen bis zum Ende der Klasse 10	8
3.4	Konkretisierung und inhaltliche Schwerpunkte bis zum Ende der Klasse 10	9
3.5	Kompetenzerwartungen bis zum Ende der Qualifikationsphase	11
3.6	Konkretisierung und inhaltliche Schwerpunkte bis zum Ende der Qualifikationsphase	12
4	Lernerfolgsüberprüfung und Leistungsbewertung.....	15
4.1	Beurteilungsbereich „Klausuren“	15
4.2	Beurteilungsbereich „Sonstige Mitarbeit“	15
4.3	Überprüfungsformen	16
5	Operatorenliste für Prüfungsaufgaben	17
6	Umsetzungsbeispiele	19

1 Vorwort

Das vorliegende Schulcurriculum der Deutschen Schule Lissabon (DSL) für Informatik ist eine Adaption des Kernlehrplans¹ für Informatik des Landes Nordrhein-Westfalen. Die dort formulierten Inhaltsfelder sind in hohem Maße deckungsgleich mit dem seit 2005 gültigen und umgesetzten Lehrplan für Informatik der DSL, der jedoch keine Kompetenzerwartungen formulierte. Diese Überarbeitung stellt nun die übergeordneten Kompetenzbereiche voran und formuliert in Verbindung mit den Inhaltsfeldern konkretisierte Kompetenzerwartungen. Auf Beschluss der Fachkonferenz vom 27.10.2017 wird der DSL-Lehrplan von 2005 durch dieses Schulcurriculum Informatik ersetzt.

1.1 Rahmenbedingungen an der DSL

Informatik wird an der DSL bereits ab Klasse 6 mit einer Wochenstunde unterrichtet. Der Unterricht in der Sekundarstufe I ist verpflichtend für alle Schüler und konzentriert sich auf die „Informationstechnische Grundbildung“ (ITG). Das dafür verwendete Curriculum entspricht dem des Europäischen Computerführerschein (ECDL) und den Schülerinnen und Schülern wird die Gelegenheit gegeben, die notwendigen Modulprüfungen abzulegen. Die DSL ist ein von der Dienstleistungsgesellschaft für Informatik (DLGI) akkreditiertes Prüfungszentrum für den ECDL.

In der Sekundarstufe II ist Informatik Wahlfach und kann von den Schülerinnen und Schülern als Alternative zu den übrigen Wahlfächern des mathematisch-naturwissenschaftlich-technischen Aufgabenfeld gewählt werden (Biologie, Chemie, Physik). Die Anzahl der Schüler, die sich in Anbetracht der Alternativen (inklusive der in portugiesischer Sprache unterrichteten Zusatzfächer) und der sich daraus ergebenden Konsequenzen für den individuellen Stundenplan für die Wahl des Fachs Informatik entschieden haben, war in der Vergangenheit so klein, dass regelmäßig in Jahrgang 11 und 12 in jahrgangsübergreifenden Kursen gearbeitet werden musste. Darum verzichtet dieses Schulcurriculum auf die Zuweisung bestimmter Inhalte in Jahrgang 11 und 12. Die Inhalte können vielmehr alternierend unterrichtet werden, z.B. Datenbanken und Automatentheorie im Wechsel mit rekursiven Datenstrukturen und Kryptologie. Die jeweiligen Semesterinhalte sind dann so zu wählen, dass den Schülern durch die Abfolge kein Nachteil entsteht.

Informatik wird an der DS Lissabon auf grundlegendem Niveau unterrichtet. Alle Kompetenzerwartungen sind dahingehend zu interpretieren.

In Kapitel 5 ist ein konkretes Beispiel der Umsetzung gegeben, das als Information für neue Kollegen, aber auch für Schüler und Eltern einen leichteren Einstieg ermöglicht. Die beschriebenen Lernsequenzen umfassen jeweils den Unterricht eines Halbjahres bzw. Semesters (vgl. Titel).

In jedem Halbjahr wird den Schülern ebenfalls die Teilnahme an einem Wettbewerb angeboten: Im Herbst am Informatik-Biber, im Frühjahr am Jugendwettbewerb Informatik. In der Jahrgangsstufe 11/12 bietet sich weiterhin die Teilnahme am Bundeswettbewerb Informatik an, sofern das Interesse der Schüler eine eigenständige Bearbeitung außerhalb des Unterrichts ermöglicht. Gefördert werden kann dies durch eine Gruppenarbeit von 90 Minuten als Einstieg (Schüler präsentieren sich gegenseitig die fünf Aufgaben und ihre Lösungsideen) und ggf. eine Präsentation der fertiggestellten Programme für die 1. Runde im Unterricht (vor dem Einsendeschluss ggf. mit Feedback durch andere Schüler zu Verständlichkeit und Vollständigkeit).

¹ vgl. <http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/informatik/informatik-klp/kernlehrplan-informatik-uebersicht.html> (18.02.2017)

2 Aufgaben und Ziele des Fachs Informatik

2.1 Fächer des mathematisch-naturwissenschaftlich-technischen Aufgabenfeld

Gegenstand der Fächer im mathematisch-naturwissenschaftlich-technischen Aufgabenfeld (III) sind die empirisch erfassbare, die in formalen Strukturen beschreibbare und die durch Technik gestaltbare Wirklichkeit sowie die Verfahrens- und Erkenntnisweisen, die ihrer Erschließung und Gestaltung dienen.

Innerhalb der von allen Fächern zu erfüllenden Querschnittsaufgaben tragen insbesondere auch die Fächer des mathematisch-naturwissenschaftlich-technischen Aufgabenfeldes im Rahmen der Entwicklung von Gestaltungskompetenz zur kritischen Reflexion geschlechter- und kulturstereotyper Zuordnungen, zur Werteerziehung, zur Empathie und Solidarität, zum Aufbau sozialer Verantwortung, zur Gestaltung einer demokratischen Gesellschaft, zur Sicherung der natürlichen Lebensgrundlagen, auch für kommende Generationen im Sinne einer nachhaltigen Entwicklung, und zur kulturellen Mitgestaltung bei. Darüber hinaus leisten sie einen Beitrag zur interkulturellen Verständigung, zur interdisziplinären Verknüpfung von Kompetenzen, auch mit gesellschaftswissenschaftlichen und sprachlich-literarisch-künstlerischen Feldern, sowie zur Vorbereitung auf Ausbildung, Studium, Arbeit und Beruf.

2.2 Das Fach Informatik

Wie kaum eine andere Fachdisziplin durchdringt die Informatik mit den von ihr entwickelten Systemen für jedermann wahrnehmbar nahezu alle Bereiche von Wirtschaft, Gesellschaft, Arbeit und Freizeit. Sie besitzt einen großen Anteil am Entwicklungsstand unserer technisierten und globalisierten Welt. Prozessorgesteuerte Geräte, Softwareprodukte und durch deren Einsatz bestimmte Verfahrensweisen und Prozesse beeinflussen und verändern unser Leben mit hoher Dynamik. Die Informatik stellt Prinzipien und Methoden zur Erforschung komplexer Phänomene und für die Entwicklung komplexer Systeme bereit, die zahlreiche andere Fachdisziplinen aufgreifen und adaptieren. Daher ist die Informatik in hohem Maße interdisziplinär ausgerichtet. Die Auseinandersetzung mit Themen und Methoden der Informatik in der Schule dient somit der Lebensvorbereitung und Orientierung in einer von der Informationstechnologie geprägten Welt.

Der Informatikunterricht der gymnasialen Oberstufe geht deutlich über eine Grundbildung im Bereich der Informations- und Kommunikationstechnologien in der Sekundarstufe I hinaus. Die Schülerinnen und Schüler erwerben über rezeptive Medienanwendungen und die interaktive Nutzung von Medienangeboten in den Fächern hinaus Fähigkeiten zur kritischen und verantwortungsvollen Analyse, Modellierung und Implementierung komplexer Informatiksysteme. Dabei konzentriert sich der Unterricht stets auf fundamentale und zeitbeständige informatische Ideen, Konzepte und Methoden und schließt auch die Auseinandersetzung mit Fragen einer menschengerechten Gestaltung und der Sicherheit von Systemen sowie der Folgen und Wirkungen des Einsatzes von Informatiksystemen ein. Schülerinnen und Schüler werden so befähigt und motiviert, auch zukünftige Entwicklungen zu nutzen, zu verstehen, hinsichtlich ihrer Wirkungen zu beurteilen und sich aktiv an der Fortentwicklung zu beteiligen.

Ausgangspunkt im Informatikunterricht ist häufig ein Problem mit lebensweltlichem Bezug. Schülerinnen und Schüler erwerben und erweitern in der aktiven Auseinandersetzung mit komplexen Problemstellungen Kompetenzen, die sie zum selbstständigen informatischen Problemlösen befähigen. Mit der Aneignung von Strategien und Techniken zur strukturierten Zerlegung im Problemlöseprozess, zur Algorithmisierung von Abläufen sowie zur formalsprachlichen, grafischen oder symbolischen Beschreibung von Sachverhalten und Zusammenhängen erwerben die Lernenden Kompetenzen zur Bewältigung von Komplexität. Die Konstruktion eines abstrakten Modells zu einer anwendungsbezogenen Problemstellung fördert das Abstraktionsvermögen sowie kreatives und strukturelles Denken. Die Umsetzung eines informatischen Modells in ein lauffähiges Informatiksystem hat für Schülerinnen und Schüler nicht nur einen hohen Motivationswert, sondern ermöglicht ihnen auch die eigenständige Überprüfung der Angemessenheit und Wirkung des Modells im Rückbezug auf die Problemstellung. Im Unterricht lassen sich umfangreiche Informatiksysteme nur in arbeitsteiliger projektorientierter Zusammenarbeit im Team erstellen. Solche Projekte können nur gelingen, wenn die gemeinsame

Arbeit strukturiert geplant und organisiert wird. Insgesamt leistet das Fach Informatik in der gymnasialen Oberstufe damit einen wichtigen Beitrag zu einer erweiterten Allgemeinbildung und allgemeinen Studierfähigkeit der Schülerinnen und Schüler.

2.3 Methodik des Informatik-Unterricht

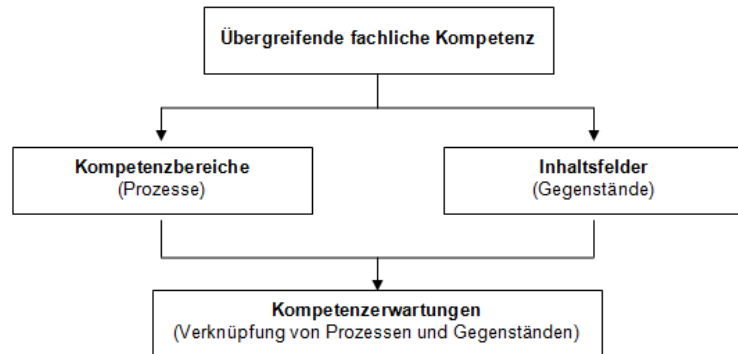
Die inhaltliche und methodische Gestaltung des Unterrichts ist entscheidend dafür, dass Schülerinnen und Schüler die ausgewiesenen Kompetenzen erwerben können. Informatikunterricht erfordert in hohem Maße die Anwendung schüleraktivierender Methoden, die selbstständiges Lernen ermöglichen und individuelle Förderung begünstigen. Unterschiedliche, auch geschlechtsspezifische Herangehensweisen, Interessen, Vorerfahrungen und fachspezifische Kenntnisse sind angemessen zu berücksichtigen.

In der Einführungsphase werden ausgehend von einfachen Fragestellungen und unter Anleitung der Lehrperson zunächst einzelne Stufen eines Problemlösungsprozesses durchlaufen. Am Ende der Qualifikationsphase sollen die Schülerinnen und Schüler dann in der Lage sein, Lösungsansätze in Form von lauffähigen Informatiksystemen weitgehend selbstständig zu entwickeln.

Der Informatikkurs der Qualifikationsphase führt unter dem Aspekt einer fundamentalen wissenschaftspropädeutischen Ausbildung in basale Sachverhalte, Probleme sowie Zusammenhänge des Faches ein und verdeutlicht die Differenz zwischen Alltagswissen und wissenschaftlich begründetem Wissen. Er fördert den Kompetenzerwerb im Bereich der wesentlichen Arbeits- und Entwicklungsmethoden der Informatik und ermöglicht das Erkennen fachlicher Zusammenhänge exemplarisch anhand geeigneter Beispiele. Ein kompletter Problemlöseprozess kann jedoch aufgrund der zeitlichen Beschränkungen nur exemplarisch erwartet werden. Die Progression der Anforderungen zwischen Einführungs- und Qualifikationsphase resultiert nicht in erster Linie aus den Prozessen selbst. Vielmehr leitet sie sich aus der zunehmenden Komplexität der fachlichen Inhalte sowie der unterrichtlichen Kontexte, Projektvorhaben und Beispiele ab, die den Schülerinnen und Schülern den Kompetenzerwerb ermöglichen.

3 Kompetenzbereiche, -erwartungen und Inhaltsfelder

Die in den allgemeinen Aufgaben und Zielen des Faches beschriebene übergreifende fachliche Kompetenz wird ausdifferenziert, indem fachspezifische Kompetenzbereiche und Inhaltsfelder identifiziert und ausgewiesen werden. Dieses analytische Vorgehen erfolgt, um die Strukturierung der fachrelevanten Prozesse einerseits sowie der Gegenstände andererseits transparent zu machen. In den Kompetenzerwartungen werden beide Seiten miteinander verknüpft. Damit wird der Tatsache Rechnung getragen, dass der gleichzeitige Einsatz von Können und Wissen bei der Bewältigung von Anforderungssituationen eine zentrale Rolle spielt.



Kompetenzbereiche repräsentieren die Grunddimensionen des fachlichen Handelns. Sie dienen dazu, die einzelnen Teiloperationen entlang der fachlichen Kerne zu strukturieren und den Zugriff für die am Lehr-Lern-Prozess Beteiligten zu verdeutlichen.

Inhaltsfelder systematisieren mit ihren jeweiligen inhaltlichen Schwerpunkten die im Unterricht der gymnasialen Oberstufe verbindlichen und unverzichtbaren Gegenstände und liefern Hinweise für die inhaltliche Ausrichtung des Lehrens und Lernens.

Kompetenzerwartungen führen Prozesse und Gegenstände zusammen und beschreiben die fachlichen Anforderungen und intendierten Lernergebnisse, die auf zwei Stufen bis zum Ende der Sekundarstufe II erreicht werden sollen.

Kompetenzerwartungen

- beziehen sich auf beobachtbare Handlungen und sind auf die Bewältigung von Anforderungssituationen ausgerichtet,
- stellen im Sinne von Regelstandards die erwarteten Kenntnisse, Fähigkeiten und Fertigkeiten auf einem mittleren Abstraktionsgrad dar,
- ermöglichen die Darstellung einer Progression vom Anfang bis zum Ende der Sekundarstufe II und zielen auf kumulatives, systematisch vernetztes Lernen,
- können in Aufgabenstellungen umgesetzt und überprüft werden.

Insgesamt ist der Unterricht in der Sekundarstufe II nicht allein auf das Erreichen der aufgeführten Kompetenzerwartungen beschränkt, sondern soll es Schülerinnen und Schülern ermöglichen, diese weiter auszubauen und darüber hinausgehende Kompetenzen zu erwerben.

Im Informatikunterricht erwerben die Lernenden eine Vielzahl von fachbezogenen Fähigkeiten und Fertigkeiten, die sich fünf Kompetenzbereichen zuordnen lassen. Diese sind nicht trennscharf, sondern haben vielfältige Bezüge zueinander. Im Sinne eines umfassenden Kompetenzerwerbs ist es daher nicht möglich, den Unterricht nach den einzelnen Kompetenzbereichen sequentiell zu strukturieren. Angemessen ist eine ganzheitliche Sicht als Grundlage für den Kompetenzerwerb.

3.1 Kompetenzbereiche

Die fachbezogenen Kompetenzen, die in Gesamtheit informatische Problemlösekompetenz ausmachen, lassen sich den fünf Kompetenzbereichen, Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren sowie Kommunizieren und Kooperieren zuordnen.

3.1.1 **Argumentieren**

Informatische Zusammenhänge, Vorgehensweisen, Lösungsansätze und Entwurfsentscheidungen bedürfen der Erläuterung und Begründung, um Transparenz,

Nachvollziehbarkeit und Überprüfbarkeit im Diskurs zu gewährleisten. Argumentieren umfasst das Erläutern, Begründen und Beurteilen in informatischen Sachzusammenhängen und Prozessen. Erläutern bedeutet, einen Sachverhalt zu veranschaulichen und verständlich zu machen. Unter Begründen wird die Darlegung von rational nachvollziehbaren Argumenten auf der Grundlage von Begriffen, Regeln, Methoden und Verfahren der Informatik verstanden. Dazu gehört auch, den Begründungszusammenhang durch geeignete Beispiele zu veranschaulichen. Beurteilen meint, zu einem informatischen Sachverhalt oder Prozess ein selbstständiges Urteil unter Verwendung von Fachwissen und Fachmethoden zu formulieren und zu begründen. Argumentieren umfasst auch die Bewertung von Nutzen, Grenzen und Auswirkungen von Informatiksystemen.

3.1.2 Modellieren

Um ein Problem aus einem inner- oder außerinformatischen Kontext lösen zu können, wird in der Regel zunächst ein informatisches Modell entwickelt, das auf einem prozessorgesteuerten Gerät implementiert werden kann. Informatisches Modellieren zielt auf eine abstrahierende Beschreibung der wesentlichen Komponenten und Parameter eines realen oder geplanten Systems sowie des Ordnungsgefüges und der Wirkungsbeziehungen zwischen ihnen. Der Modellierungsprozess beginnt mit der Analyse und einer strukturierten Zerlegung des Ausgangsproblems. Teilkomponenten müssen identifiziert, konstruiert und gegebenenfalls miteinander vernetzt werden. Ein Ergebnis eines Modellierungsprozesses ist in der Regel eine formale, textuelle oder grafische Darstellung.

3.1.3 Implementieren

Implementieren umfasst die Umsetzung eines Modells in ein Informatiksystem. Dazu gehören das Programmieren, Evaluieren und Validieren von Modellbestandteilen unter Nutzung geeigneter Werkzeuge. Grundlegende Methoden und Denkweisen der Programmentwicklung werden dabei in den Vordergrund gestellt. Die Programmerstellung ist ein bedeutsamer Bestandteil des Problemlösungsprozesses, weil erst dadurch das Modell wirksam wird. An dem entstandenen Informatiksystem können Wirkungen der Modellentscheidungen diskutiert sowie Ursachen und Tragweite von möglichen Fehlern im Modell erkannt und korrigiert werden. Dadurch werden die Selbstreflexion des Lösungsprozesses und eine vertiefte Modellkritik unterstützt.

3.1.4 Darstellen und Interpretieren

Die Informatik hat zur Unterstützung von Problemlöse- und Modellbildungsprozessen ein reiches Repertoire an Darstellungsformen entwickelt. Schülerinnen und Schüler werden nach und nach mit unterschiedlichen Darstellungsformen konfrontiert, die sie in inner- und außerinformatischen Kontexten selbst nutzen. Vorgegebene Darstellungen müssen anwendungsbezogen interpretiert werden. Im Rahmen eigener Problemlösungen müssen angemessene Darstellungsformen unter Verwendung der fachspezifischen Notation angewendet werden. Dies fördert ein Verständnis von Zusammenhängen und Bezügen zwischen unterschiedlichen informatischen Sachverhalten sowie die Fähigkeit, diese anderen deutlich zu machen.

3.1.5 Kommunizieren und Kooperieren

Die Kenntnis und Nutzung arbeitsteiliger und kooperativer Vorgehensweisen ist für die Entwicklung komplexer Informatiksysteme erforderlich, um prozessorientiertes Arbeiten zu planen und abzusichern. Zum Kommunizieren im Sinne eines fachlichen Austausches gehören die sachadäquate Darstellung und Dokumentation zur Weitergabe von Sachverhalten sowie die Nutzung geeigneter Werkzeuge, die die Kommunikation unterstützen. Für eine sachangemessene und präzise Verständigung über informatische Gegenstände sind ein angemessener Umgang mit Fachbegriffen und der sukzessive Aufbau einer Fachsprache unerlässlich.

3.2 Inhaltsfelder

Kompetenzerwerb ist an fachliche Inhalte gebunden. Die für den Informatikunterricht obligatorischen Inhalte, an denen die Kompetenzen entwickelt werden sollen, lassen sich den

folgenden fünf Inhaltsfeldern zuordnen. Diese werden bereits in der Einführungsphase aufgegriffen und in der Qualifikationsphase vertieft.

3.2.1 Daten und ihre Strukturierung

Die automatische Verarbeitung von Informationen mittels Maschinen ist überhaupt erst durch deren digitale Repräsentation in Form von Daten möglich. Für die rechnergestützte Lösung von Problemen in inner- und außerinformatischen Kontexten müssen daher Informationen in angemessener Struktur durch Daten und zugehörige Operationen repräsentiert werden, so dass die Daten zielgerichtet und effizient automatisch verarbeitet und die Ergebnisse wiederum als Information interpretiert werden können.

3.2.2 Algorithmen

Zu vielen bedeutenden wissenschaftlichen Erfolgen und technischen Errungenschaften der jüngeren Zeit hat die Informatik maßgeblich beigetragen. Neben der rasanten Steigerung der Leistungsfähigkeit der technischen Systeme sind diese Fortschritte insbesondere der Entwicklung von innovativen und effizienten Algorithmen zu verdanken. Ein Algorithmus ist eine genaue Beschreibung von Handlungsschritten zur Lösung eines Problems, die von einem Prozessor ausgeführt werden können. Häufig verwendete Grundkonstrukte von Algorithmen sowie Algorithmen, die im Kontext bestimmter Problemklassen von elementarer Bedeutung sind, lassen sich unter Berücksichtigung ihrer Effizienz adaptieren, um neue Aufgabenstellungen in konkreten Anwendungskontexten problemgerecht einer automatischen Verarbeitung zuzuführen.

3.2.3 Formale Sprachen und Automaten

Der Einsatz von Informatiksystemen zur Lösung komplexer Probleme ist nur unter Verwendung formaler Sprachen als Mittler zwischen Mensch und Maschine möglich. Sprachen dienen zur Kommunikation und genügen Regeln zur Bildung von Wörtern und Sätzen. Formale Sprachen der Informatik werden durch Grammatiken präzise beschrieben. Zu formalen Sprachen können Automaten entwickelt werden, die die Wörter der Sprache akzeptieren oder weiterverarbeiten. Eine fachliche Beschreibung von Automaten mithilfe einer Menge von Zuständen samt Regeln für die zeitliche Abfolge von Zustandsübergängen ist als Modellierungstechnik in verschiedenen Problemfeldern anwendbar. Automaten eignen sich in besonderem Maße, um mithilfe theoretischer Betrachtungen auch die Grenzen von Automatenmodellen zu beleuchten.

3.2.4 Informatiksysteme

Informatiksysteme sind heute weltweit miteinander vernetzt. Ein Informatiksystem ist eine spezifische Zusammenstellung von Hardware-, Software- und Netzwerkkomponenten zur Lösung eines Anwenderproblems. Gegenstand der Betrachtung in diesem Inhaltsfeld sind schwerpunktmäßig der prinzipielle Aufbau singularer und vernetzter Rechnersysteme und deren Interaktion untereinander und mit dem Benutzer.

3.2.5 Informatik, Mensch und Gesellschaft

Informatiksysteme stehen in intensiver Wechselwirkung mit Individuum und Gesellschaft. Ihr Einsatz hat weitreichende Konsequenzen für unsere Lebens- und Arbeitswelt. Handlungsspielräume müssen im Spannungsfeld von Rechten und Interessen des Individuums, gesellschaftlicher Verantwortung und möglichen Sicherheitsrisiken wahrgenommen werden.

3.3 Kompetenzerwartungen bis zum Ende der Klasse 10

Der Unterricht soll es den Schülerinnen und Schülern ermöglichen, dass sie – aufbauend auf einer ggf. heterogenen Kompetenzentwicklung in der Sekundarstufe I – am Ende der Einführungsphase (=Klasse 10) über die im Folgenden genannten Kompetenzen verfügen. Dabei werden zunächst übergeordnete Kompetenzerwartungen zu allen Kompetenzbereichen aufgeführt und im Anschluss zusätzlich inhaltsfeldbezogen konkretisiert. Bei einigen Kompetenzerwartungen ist eine Progression zwischen Einführungs- und Qualifikationsphase nicht explizit ausgewiesen. Ein Kompetenzzuwachs wird aber gleichwohl erwartet. Dieser ergibt sich immer auch aus

- der Zugänglichkeit der Kontexte und Beispiele,
- dem Umfang und der Komplexität der Problemstellungen sowie der Lösungswege,

- den Anforderungen, die an das Abstraktionsvermögen und das analytische Denken gestellt werden,
- dem zur Problemlösung erforderlichen Grad des Verfügens über Problemlösestrategien,
- dem Grad der Vernetzung der Kompetenzen, die für eine Bearbeitung von Problemstellungen erforderlich sind,
- dem Grad der Selbstständigkeit der Schülerinnen und Schüler bei der Problemlösung,
- dem Grad der Arbeitsteilung bei der Problemlösung.

3.3.1 ARGUMENTIEREN (A)

Die Schülerinnen und Schüler

- erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme,
- analysieren und erläutern informatische Modelle,
- analysieren und erläutern Computerprogramme,
- beurteilen die Angemessenheit informatischer Modelle.

3.3.2 MODELLIEREN (M)

Die Schülerinnen und Schüler

- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle,
- modifizieren und erweitern informatische Modelle.

3.3.3 IMPLEMENTIEREN (I)

Die Schülerinnen und Schüler

- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme,
- modifizieren und erweitern Computerprogramme,
- testen und korrigieren Computerprogramme.

3.3.4 DARSTELLEN UND INTERPRETIEREN (D)

Die Schülerinnen und Schüler

- interpretieren Daten und erläutern Beziehungen und Abläufe, die in Form von textuellen und grafischen Darstellungen gegeben sind,
- überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform,
- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar.

3.3.5 KOMMUNIZIEREN UND KOOPERIEREN (K)

Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte,
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit,
- präsentieren Arbeitsabläufe und -ergebnisse.

3.4 Konkretisierung und inhaltliche Schwerpunkte bis zum Ende der Klasse 10

Verknüpft man die Kompetenzerwartungen mit den inhaltlichen Schwerpunkten, so ergeben sich die nachfolgenden konkretisierten Kompetenzerwartungen. In der Regel lässt sich eine konkretisierte Kompetenzerwartung nicht eindeutig einem Kompetenzbereich zuordnen. Der in Klammern hinter einer konkretisierten Kompetenzerwartung angegebene Kompetenzbereich weist jeweils den stärksten Bezug zu ihr auf. Die inhaltsbezogenen Kompetenzen sind darüber hinaus mit weiteren übergeordneten Kompetenzen verknüpft, insbesondere mit denen aus dem Bereich „Kommunizieren und Kooperieren“.

3.4.1 Inhaltsfeld 1: Daten und ihre Strukturierung

Schwerpunkt: Objekte und Klassen

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),
- modellieren Klassen unter Verwendung von Vererbung (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),
- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),
- stellen den Zustand eines Objekts dar (D),
- stellen die Kommunikation zwischen Objekten grafisch dar (M),
- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),
- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D),
- analysieren und erläutern eine objektorientierte Modellierung (A),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).

3.4.2 Inhaltsfeld 2: Algorithmen

Schwerpunkt: Entwurf und Implementierung einfacher Algorithmen zum Suchen und Sortieren

Die Schülerinnen und Schüler

- analysieren und erläutern einfache Algorithmen und Programme (A),
- modifizieren einfache Algorithmen und Programme (I),
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- testen Programme schrittweise anhand von Beispielen (I),
- analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D),
- entwerfen einen weiteren Algorithmus zum Sortieren (M),
- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf (A).

3.4.3 Inhaltsfeld 3: Formale Sprachen und Automaten

Schwerpunkt: Syntax und Semantik einer Programmiersprache

Die Schülerinnen und Schüler

- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

3.4.4 Inhaltsfeld 4: Informatiksysteme

Schwerpunkt: lokales Dateisystem und Internet

Die Schülerinnen und Schüler

- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K),
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).

3.4.5 Inhaltsfeld 5: Informatik, Mensch und Gesellschaft

Schwerpunkt: Einsatz von Informatiksystemen

Die Schülerinnen und Schüler

- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).

- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).

3.5 Kompetenzerwartungen bis zum Ende der Qualifikationsphase

Der Unterricht soll es den Schülerinnen und Schülern ermöglichen, dass sie – aufbauend auf der Kompetenzentwicklung in der Einführungsphase – am Ende der Sekundarstufe II über die im Folgenden genannten Kompetenzen verfügen.

3.5.1 ARGUMENTIEREN (A)

Die Schülerinnen und Schüler

- erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme,
- zeigen im Problemlösungsprozess Alternativen auf und begründen ihre Auswahlentscheidungen,
- analysieren und erläutern informatische Modelle,
- analysieren und erläutern Computerprogramme,
- beurteilen die Angemessenheit von Modellierungen und Implementationen,
- erläutern und beurteilen informatische Modelle und Informatiksysteme hinsichtlich ihrer Möglichkeiten, Grenzen und Auswirkungen.

3.5.2 MODELLIEREN (M)

Die Schülerinnen und Schüler

- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle,
- modifizieren und erweitern informatische Modelle,
- wenden im Modellierungsprozess geeignete Lösungsstrategien an.

3.5.3 IMPLEMENTIEREN (I)

Die Schülerinnen und Schüler

- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme,
- modifizieren und erweitern Computerprogramme,
- testen und korrigieren Computerprogramme systematisch.

3.5.4 DARSTELLEN UND INTERPRETIEREN (D)

Die Schülerinnen und Schüler

- interpretieren Daten und erläutern Beziehungen und Abläufe, die in Form von textuellen, grafischen oder formalen Darstellungen gegeben sind,
- überführen gegebene textuelle, grafische oder formale Darstellungen informatischer Zusammenhänge in eine der anderen Darstellungsformen,
- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen, Grafiken und Formalismen dar.

3.5.5 KOMMUNIZIEREN UND KOOPERIEREN (K)

Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte,
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten,
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen,
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse,
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht.

3.6 Konkretisierung und inhaltliche Schwerpunkte bis zum Ende der Qualifikationsphase

Verknüpft man die Kompetenzerwartungen mit den inhaltlichen Schwerpunkten, so ergeben sich die nachfolgenden konkretisierten Kompetenzerwartungen. In der Regel lässt sich eine konkretisierte Kompetenzerwartung nicht eindeutig einem Kompetenzbereich zuordnen. Der in Klammern hinter einer konkretisierten Kompetenzerwartung angegebene Kompetenzbereich weist jeweils den stärksten Bezug zu ihr auf. Die inhaltsbezogenen Kompetenzen sind darüber hinaus mit weiteren übergeordneten Kompetenzen verknüpft, insbesondere mit denen aus dem Bereich „Kommunizieren und Kooperieren“.

3.6.1 Inhaltsfeld 1: Daten und ihre Strukturierung

Schwerpunkt: Objekte und Klassen

Die Schülerinnen und Schüler

- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),
- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen (M),
- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),
- verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M),
- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),
- stellen die Kommunikation zwischen Objekten grafisch dar (D),
- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),
- dokumentieren Klassen (D),
- analysieren und erläutern objektorientierte Modellierungen (A),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).

Schwerpunkt: Datenbanken

Die Schülerinnen und Schüler

- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),
- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D),
- modifizieren eine Datenbankmodellierung (M),
- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),
- bestimmen Primär- und Sekundärschlüssel (M),
- analysieren und erläutern eine Datenbankmodellierung (A),
- erläutern die Eigenschaften normalisierter Datenbankschemata (A),
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).

3.6.2 Inhaltsfeld 2: Algorithmen

Schwerpunkt: Analyse, Entwurf und Implementierung von Algorithmen

Die Schülerinnen und Schüler

- analysieren und erläutern Algorithmen und Programme (A),
- modifizieren Algorithmen und Programme (I),
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),

- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),
- testen Programme systematisch anhand von Beispielen (I).

Schwerpunkt: Algorithmen in ausgewählten informatischen Kontexten

Die Schülerinnen und Schüler

- erläutern Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (A),
- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),
- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),
- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D).

3.6.3 Inhaltsfeld 3: Formale Sprachen und Automaten

Schwerpunkt: Syntax und Semantik einer Programmiersprache

Die Schülerinnen und Schüler

- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),
- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I).

Schwerpunkt: Endliche Automaten

Die Schülerinnen und Schüler

- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A),
- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),
- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),
- zeigen die Grenzen endlicher Automaten im Anwendungszusammenhang auf (A).

Schwerpunkt: Grammatiken regulärer Sprachen

Die Schülerinnen und Schüler

- analysieren und erläutern Grammatiken regulärer Sprachen (A),
- modifizieren Grammatiken regulärer Sprachen (M),
- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),
- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),
- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M),
- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).
- zeigen die Grenzen regulärer Grammatiken im Anwendungszusammenhang auf (A).

3.6.4 Inhaltsfeld 4: Informatiksysteme

Schwerpunkt: Einzelrechner und Rechnernetzwerke

Die Schülerinnen und Schüler

- erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A),

- beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A).

Schwerpunkt: Einzelrechner und Rechnernetzwerke

Die Schülerinnen und Schüler

- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I).

Schwerpunkt: Datensicherheit

Die Schülerinnen und Schüler

- erläutern Eigenschaften, Funktionsweisen und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),
- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).

3.6.5 Inhaltsfeld 5: Informatik, Mensch und Gesellschaft

Schwerpunkt: Automatisierung

Die Schülerinnen und Schüler

- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),
- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),
- untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).

4 Lernerfolgsüberprüfung und Leistungsbewertung

Erfolgreiches Lernen ist kumulativ. Entsprechend sind die Kompetenzerwartungen in der Regel in ansteigender Progression und Komplexität formuliert. Dies erfordert, dass Lernerfolgsüberprüfungen darauf ausgerichtet sein müssen, Schülerinnen und Schülern Gelegenheit zu geben, Kompetenzen, die sie in den vorangegangenen Jahren erworben haben, wiederholt und in wechselnden Zusammenhängen unter Beweis zu stellen. Für Lehrerinnen und Lehrer sind die Ergebnisse der begleitenden Diagnose und Evaluation des Lernprozesses sowie des Kompetenzerwerbs Anlass, die Zielsetzungen und die Methoden ihres Unterrichts zu überprüfen und ggf. zu modifizieren. Für die Schülerinnen und Schüler sollen ein den Lernprozess begleitendes Feedback sowie Rückmeldungen zu den erreichten Lernständen eine Hilfe für die Selbsteinschätzung sowie eine Ermutigung für das weitere Lernen darstellen. Die Beurteilung von Leistungen soll demnach grundsätzlich mit der Diagnose des erreichten Lernstandes und Hinweisen zum individuellen Lernfortschritt verknüpft sein.

Die Leistungsbewertung ist so anzulegen, dass sie den in den Fachkonferenzen beschlossenen Grundsätzen entspricht und dass die Kriterien für die Notengebung den Schülerinnen und Schülern transparent sind. Im Sinne der Orientierung an den zuvor formulierten Anforderungen sind grundsätzlich alle Kompetenzbereiche („Argumentieren“, „Modellieren“, „Implementieren“, „Darstellen und Interpretieren“ und „Kommunizieren und Kooperieren“) bei der Leistungsbewertung angemessen zu berücksichtigen. Überprüfungsformen schriftlicher, mündlicher und ggf. praktischer Art sollen deshalb darauf ausgerichtet sein, die Erreichung der dort aufgeführten Kompetenzerwartungen zu überprüfen. Ein isoliertes, lediglich auf Reproduktion angelegtes Abfragen einzelner Daten und Sachverhalte allein kann dabei den zuvor formulierten Ansprüchen an die Leistungsfeststellung nicht gerecht werden.

4.1 Beurteilungsbereich „Klausuren“

Für den Einsatz in Klausuren kommen im Wesentlichen Überprüfungsformen – ggf. auch in Kombination – in Betracht, die im letzten Abschnitt dieses Kapitels aufgeführt sind. Die Schülerinnen und Schüler müssen mit den Überprüfungsformen, die im Rahmen von Klausuren eingesetzt werden, vertraut sein und rechtzeitig sowie hinreichend Gelegenheit zur Anwendung haben. Da in Klausuren neben der Verdeutlichung des fachlichen Verständnisses auch die Darstellung bedeutsam ist, muss diesem Sachverhalt bei der Leistungsbewertung hinreichend Rechnung getragen werden.

4.2 Beurteilungsbereich „Sonstige Mitarbeit“

Im Beurteilungsbereich „Sonstige Mitarbeit“ können – neben den nachfolgend aufgeführten Überprüfungsformen – vielfältige weitere zum Einsatz kommen, für die kein abschließender Katalog festgesetzt ist. Im Rahmen der Leistungsbewertung gelten auch für diese die oben ausgeführten allgemeinen Ansprüche der Lernerfolgsüberprüfung und Leistungsbewertung. Im Verlauf der gymnasialen Oberstufe ist sicherzustellen, dass Formen, die im Rahmen der mündlichen Abiturprüfung von Bedeutung sind, frühzeitig vorbereitet und angewendet werden.

Zu den Bestandteilen der „Sonstigen Mitarbeit“ zählen u. a. unterschiedliche Formen der selbstständigen und kooperativen Aufgabenerfüllung, Beiträge zum Unterricht, von der Lehrkraft abgerufene Leistungsnachweise wie z.B. die schriftliche Übung, von der Schülerin oder dem Schüler vorbereitete, in abgeschlossener Form eingebrachte Elemente zur Unterrichtsarbeit, die z.B. in Form von Präsentationen, Protokollen, Referaten und Portfolios möglich werden. Schülerinnen und Schüler bekommen durch die Verwendung einer Vielzahl von unterschiedlichen Überprüfungsformen vielfältige Möglichkeiten, ihre eigene Kompetenzentwicklung darzustellen und zu dokumentieren.

Der Bewertungsbereich „Sonstige Mitarbeit“ erfasst die im Unterrichtsgeschehen durch mündliche, schriftliche und ggf. praktische Beiträge sichtbare Kompetenzentwicklung der Schülerinnen und Schüler. Der Stand der Kompetenzentwicklung in der „Sonstigen Mitarbeit“ wird sowohl durch Beobachtung während des Schuljahres (Prozess der Kompetenzentwicklung) als auch durch punktuelle Überprüfungen (Stand der Kompetenzentwicklung) festgestellt.

4.3 Überprüfungsformen

Die Kompetenzerwartungen ermöglichen eine Vielzahl von Überprüfungsformen. Im Verlauf der gesamten gymnasialen Oberstufe soll – auch mit Blick auf die individuelle Förderung – ein möglichst breites Spektrum der genannten Formen in schriftlichen, mündlichen oder praktischen Kontexten zum Einsatz gebracht werden. Darüber hinaus können weitere Überprüfungsformen nach Entscheidung der Lehrkraft eingesetzt werden. Wichtig für die Nutzung der Überprüfungsformen im Rahmen der Leistungsbewertung ist es, dass sich die Schülerinnen und Schüler zuvor im Rahmen von Anwendungssituationen hinreichend mit diesen vertraut machen konnten. Weitere über die Auflistung hinausgehende Überprüfungsformen sind möglich.

- a) Analyse und Eingrenzung einer kontextbezogenen Problemstellung und Entwicklung eines Modells oder Teilmodells mit erläuternden Begründungen der Entwurfsentscheidungen
- b) Analyse, Erläuterung und Modifikation eines vorgegebenen informatischen Modells sowie die vergleichende Beurteilung unterschiedlicher Entwürfe
- c) Vollständige oder teilweise Implementation einer bereits modellierten Problemstellung
- d) Entwurf und formale Darstellung von Algorithmen zu einer vorgegebenen informatischen Problemstellung
- e) Analyse und Erläuterung von vorgegebenen Algorithmen oder Programmausschnitten
- f) Interpretation gegebener textueller, grafischer oder formaler Darstellungen informatischer Zusammenhänge und deren Überführung in eine andere Darstellungsform
- g) Darstellung, Erläuterung und sachgerechte Anwendung von informatischen Begriffen, Verfahren und Lösungsstrategien
- h) Analyse und Beurteilung einer Problemlösung oder eines Informatiksystems nach vorgegebenen oder eigenen Kriterien
- i) Analyse und Bewertung des Einsatzes eines Informatiksystems in Bezug auf ethische, rechtliche oder gesellschaftliche Fragestellungen

5 Operatorenliste für Prüfungsaufgaben

Für Aufgabenstellungen in schriftlichen und mündlichen Prüfungen bieten sich die folgenden Operatoren² an, die wie nachfolgend dargestellt den Anforderungsbereichen (AFB) zugeordnet sind:

Operator	Definition	AFB
analysieren	eine konkrete Materialgrundlage untersuchen, einzelne Elemente identifizieren und Beziehungen zwischen den Elementen erfassen; „analysieren“ wird immer in Kombination mit einem weiteren Operator benutzt, der angibt, wie das Analyseergebnis darzustellen ist	II-III
angeben	ohne nähere Erläuterungen und Begründungen aufzählen, nennen	I
anwenden	ein bekanntes Verfahren in einer neuen Situation verwenden; begründen „anwenden“ wird häufig in Kombination mit einem weiteren Operator verwendet	II
begründen	einen Sachverhalt oder eine Entwurfsentscheidung durch Angabe von Gründen erklären	II-III
beschreiben	Sachverhalte oder Zusammenhänge unter Verwendung der Fachsprache in eigenen Worten verständlich wiedergeben	II
bestimmen	mittels charakteristischer Merkmale einen Sachverhalt genau feststellen und beschreiben	II
beurteilen	zu einem Sachverhalt ein selbstständiges Urteil unter Verwendung von Fachwissen und Fachmethoden formulieren und begründen	III
darstellen	Zusammenhänge oder Sachverhalte in strukturierter Form graphisch oder sprachlich wiedergeben	I-II
dokumentieren	Arbeitsergebnisse oder Arbeitsverfahren in strukturierter Form wiedergeben	I-II
entscheiden	sich begründet bei vorgegebenen Alternativen auf eine Möglichkeit festlegen	II
entwerfen	herstellen und gestalten eines Systems von Elementen unter vorgegebener Zielsetzung	II-III
entwickeln	herstellen und gestalten eines Systems von Elementen unter vorgegebener Zielsetzung	II-III
erläutern	einen Sachverhalt auf der Grundlage von Vorkenntnissen so darlegen, dass er verständlich wird	I-II
ermitteln	mittels charakteristischer Merkmale einen Sachverhalt genau feststellen und beschreiben	II
erweitern	eine gegebene Struktur gemäß konkreter Vorgaben ergänzen	II
implementieren	umsetzen eines informatischen Modells oder Algorithmus' in eine Programmiersprache	II-III
interpretieren	Sinnzusammenhänge aus Materialien erschließen	II-III

² Quelle: <https://www.standardsicherung.schulministerium.nrw.de/cms/zentralabitur-gost/faecher/getfile.php?file=3939> (01.03.18)

Operator	Definition	AFB
modellieren	zu einem Ausschnitt der Realität ein informatisches Modell anfertigen	II-III
modifizieren	eine gegebene Struktur gemäß konkreter Vorgaben verändern	II
Stellung nehmen	unter Heranziehung relevanter Sachverhalte die eigene Meinung zu einem Problem argumentativ entwickeln und darlegen	III
überführen	eine Darstellung in eine andere Darstellungsform bringen	I-II
vergleichen	nach vorgegebenen oder selbst gewählten Kriterien Gemeinsamkeiten, Ähnlichkeiten und Unterschiede ermitteln und darstellen	II
vervollständigen	eine gegebene Struktur gemäß konkreter Vorgaben erweitern oder verändern	II
zeigen	eine Aussage, einen Sachverhalt nach Berechnungen, Herleitungen oder logischen Begründungen bestätigen	II

6 Umsetzungsbeispiele

Die folgenden Ablauf-Pläne sind Umsetzungsbeispiele für das Schulcurriculum, auf dessen Basis die schuleigen Moodle-Kurse ausgerichtet sind. In diesen Moodlekursen stehen Lernmaterialien und Aufgabenstellungen zur Verfügung, die ein konkretes Lehrbuch ersetzen können, inkl. vieler Verlinkungen auf Lernmaterialien oder Informationsquellen im Internet.

Dieser Ablaufplan ist auf Grundlage des alten Rahmenplans entstanden, der das Curriculum in der Programmiersprache Delphi umsetzte. 2014-2017 musste konzeptionell mit jahrgangsübergreifendem Unterricht in Klasse 11 und 12 gearbeitet werden, was Einschränkungen in der Abfolge bestimmter Themen mit sich bringt. Die Aufnahme dieses Umsetzungsbeispiels in das Schulcurriculum soll einerseits die Einarbeitung neuer Kollegen an der DSL erleichtern und andererseits das Curriculum für Schüler und Eltern verständlicher machen. Weitere Umsetzungsbeispiele können angefügt werden, die letzten zwei müssen von der nächsten Fachkonferenz überarbeitet werden (vor der nächsten Umsetzung).

Informatik Jahrgangsstufe 10.1		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Inhaltsfeldern
<p>Einführung in die Programmierung (mit Java)</p> <ul style="list-style-type: none"> • Geschichte der (höheren) Programmiersprachen, Entwicklung von Java, Unterscheidung in Compiler und Interpreter, Quelltext und Class-Datei, Applet und Standalone, Aufbau eines Java-Programms: Hallo-World • Eingaben mittels importierter Programmteile, Ausgaben auf die Konsole, Deklaration von Variablen, Zuweisungen und Vergleiche • Java API • Bedingte Anweisungen (IF-ELSE, SWITCH) • Schleifen in Java (WHILE, DO-WHILE, FOR) • Quelltextformatierungen und Kommentare • Einfache Algorithmen und ihre grafische Darstellung in Form eines Programm-Ablauf-Plans, Nassi-Schneidermann Struktogramme • Nutzung einer Entwicklungsumgebung (BlueJ) • Verwendung von Methoden mit Parametern und Rückgabetypen • Datenstruktur: Eindimensionale Felder • Komplexität und Bewertung von Algorithmen 	<ul style="list-style-type: none"> • JRE, JDK, SDK, Installation • Compiler und Interpreter (von der Kommandozeile aus) • „Hallo World“ und Konsolenausgabe • import-Anweisung, System.out.println(), Prog1Tools.readChar(), etc. if() else() Anweisungen switch() Anweisung • TalkBots erstellen und erweitern • Variablentypen und -deklaration (int, char, float, double, boolean, String) • Zeilen-, Mehrzeilen und Dokumentationskommentare • Konventionen zur Quelltextgestaltung • Zahlenraten / Mastermind als PAP und Struktogramm, mit Zufallszahlen • BlueJ, statische Methoden Beispiel: Euklidischer Algorithmus • BlueJ, Daten als Felder, Sortieralgorithmen Beispiel: BubbleSort, InsertionSort 	<p>3.4.4</p> <p>3.4.3</p> <p>3.4.1</p> <p>3.4.2</p>

Informatik Jahrgangsstufe 10.2		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Einführung in die Objektorientierung (mit Java: Greenfoot)</p> <ul style="list-style-type: none"> • Greenfoot Entwicklungsumgebung als Hilfestellung zur Arbeit mit Objekten, Greenfoot API • Einfache Objekte und ihre Eigenschaften (World, Actor, act()) • Deklaration von Methoden und ihr Aufruf • Vererbung von Eigenschaften und Fähigkeiten • Modulierung durch Generalisierung • Verwendung von Konstruktoren • globale und lokale Variablen • statische und Objektvariablen, statische und Objektmethoden • Datenkapselung (public, protected und private) 	<ul style="list-style-type: none"> • Greenfoot, Installation, Szenarien, GreenFootGallery zum Austausch • Krabben-Szenario, addObject(), move(), turn(), atWorldEdge() • Steuerung von Objekten: ifKeyIsDown() • Kollisionen (canSee(), eat(), getOneObjectAtOffset()) <p>Für das zweite Halbjahr bietet sich die Projektarbeit an einem eigenen Spiel bzw. die Weiterentwicklung des Krabben-Szenarios (für leistungsschwächere Schüler) an.</p>	<p>3.4.3</p> <p>3.4.2</p> <p>3.4.1</p>

Informatik Jahrgangsstufe 11/12 – Kryptologie		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Zweck und historische Ansätze der Datenverschlüsselung</p> <ul style="list-style-type: none"> Grundlagen der symmetrischen Verschlüsselung <p>Moderne Kryptologie – und neue Herausforderungen</p> <ul style="list-style-type: none"> Einwegfunktionen ermöglichen Public-Key Datenbanken PGP und moderne Zertifikate Web of Trust und Man-In-The-Middle Angriffe <p style="padding-left: 40px;">Den Jugendlichen wird deutlich, dass die Sicherheit moderner Verschlüsselungsverfahren auf den praktischen Grenzen der Berechenbarkeit beruht.</p> <p>Geschäftsmodelle basierend auf BitCoins</p> <p>Missbrauchsmöglichkeiten durch „Krypto-Trojaner“</p>	<ul style="list-style-type: none"> Der Goldkäfer (nach E.A.Poe) und eine erste Kryptoanalyse mono- bis Polyalphabetische Verschlüsselungen (historische Beispiele und Ansätze) Substitution und Transposition Ansätze der Kryptoanalyse, Häufigkeitsanalyse Grenzen der symmetrischen Verschlüsselung (Kasiski-Test) Exkurs: Die Enigma (ggf. als Schülerreferat) Das Schlüsselverteilungsproblem Asymmetrische Ansätze: Public Key Verfahren (RSA, Elgamal) PGP Zertifikate und certification authority (CA) hoher Laufzeitaufwand als Schutz vor Entschlüsselung durch systematisches Ausprobieren aller Möglichkeiten (Brute-Force-Verfahren) Ransomware und Userverhalten Aktuelle Herausforderungen für den Datenschutz 	<p>3.6.4 (Datensicherheit)</p> <p>3.6.5</p>

Informatik Jahrgangsstufe 11/12 – Formale Sprachen & Automaten		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Theoretische Informatik: Formale Sprachen und Automaten</p> <p>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.</p>	<ul style="list-style-type: none"> 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, Nichtterminal, 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 	<p>3.6.3 (Automaten)</p>

Informatik Jahrgangsstufe 11/12 – Automatisierungstechnik		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Praktische Informatik - Technische Systeme</p> <p>Elektronische Wirksysteme</p> <ul style="list-style-type: none"> Anwendungsbezogene digitaltechnische Grundsaltungen und -funktionen <p>Mikrocomputertechnik</p> <ul style="list-style-type: none"> Die Schüler analysieren den Aufbau und die Arbeitsweise digitaler Computer und entwickeln hardwarenah unter Beachtung unterschiedlicher Programmierungstechniken Anwendungsprogramme. <p>Automatisierungstechnik</p> <ul style="list-style-type: none"> Die Schüler arbeiten mit speicherprogrammierbaren Steuerungen, Sensoren und Aktoren zur Programmierung einfacher Automatisierungsanlagen 	<ul style="list-style-type: none"> Die SchülerInnen werden deduktiv an elektronische und digitale Grundsaltungen inkl. Ihrer Bauelemente und physikalischen Wirkprinzipien herangeführt. Logikschaltung einer Ampelanlage Sie lernen relevante Zahlensysteme kennen. Sie können einfache elektronische Schaltungen hinsichtlich ihres Aufbaus und ihrer Wirkung analysieren und logische Verknüpfungsschaltungen unter Zuhilfenahme von Wertetabellen, Logiksymbolen und elementaren Funktionsbausteinen (z.B. Timer, RS-Flip-Flop) erstellen und fachsprachlich erläutern. Grundkomponenten des Computers, Grundaufbau der CPU (Mikroprozessor / Mikrocontroller), Programmierung in Maschinencode Programmierung in Assembler/ C / Java Assembler/Dissassembler, Debugger, AD-, DA-Wandler, Schnittstellen, Befehlstrukturen, PC-Entwicklungssystem, Algorithmen, Betriebssystemroutinen Die Schüler erarbeiten unter Beachtung beruflicher Anforderungen steuerungstechnische Anlagen (z.B. Klein-SPS- LOGO Siemens). 	<p>3.6.5 (Automatisierung)</p> <p>Technische Probleme analysieren, Wirkungszusammenhänge ermitteln, Lösungen entwickeln und deren Wirksamkeit beurteilen.</p> <p>Optimieren von Lösungen für technische Aufgabenstellungen durch Strukturieren von Lösungswegen, Feststellen möglicher Lösungsvarianten, Vergleichen der Lösungsvarianten,</p> <p>Auswählen einer Variante und Darstellen des Kompromisscharakters der bevorzugten Lösung.</p>

Informatik Jahrgangsstufe 11/12 – Rekursive Datenstrukturen 1 (Listen)		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Praktische Informatik: Listen</p> <p>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</p> <p>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.</p>	<ul style="list-style-type: none"> • Methoden der Datenstruktur Schlange: <ul style="list-style-type: none"> ○ 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: <ul style="list-style-type: none"> ○ 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 	<p>3.6.2</p>

Informatik Jahrgangsstufe 11/12 – Rekursive Datenstrukturen 2 (Bäume)		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Praktische Informatik: Bäume als spezielle Graphen</p> <p>In Erweiterung ihrer Grundkenntnisse über das hierarchische Ordnungsprinzip lernen die Schüler den Baum als effiziente dynamische Datenstruktur kennen.</p> <p>Sie stellen fest, dass damit viele Strukturen aus anderen Gebieten abgebildet werden können.</p> <p>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.</p> <p>Bei der sprachlichen bzw. graphischen Darstellung und insbesondere bei der Implementierung dieser Datenstruktur vertiefen sie ihr Verständnis für das Prinzip der Rekursion.</p> <p>Im Rahmen praktischer Fragestellungen, z. B. zur Planung von Verkehrsrouten, wenden die Schüler auch die Datenstruktur Graph als Erweiterung der Struktur Baum an.</p>	<ul style="list-style-type: none"> • 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 	<p>3.6.2</p>

Informatik Jahrgangsstufe 11/12 – (Web-) Datenbanken		
Lernsequenz	Unterrichtsinhalte	Verknüpfung mit Kompetenzbereichen
<p>Netzwerke und Netzinhalte</p> <ul style="list-style-type: none"> • Kommunikationsgrundlagen (Normen und Modelle, Topologien) • Verkabelungskonzepte, Netzwerkprotokolle, Ethernet-Frames • Client-Server Strukturen, vernetzte Rechner, Netzwerk-Grundlagen (IP, Subnet-Addressierung, LAN/WAN, Routing, DHCP, DNS, Ping, FTP, HTTP, SSH) • OSI-Referenzmodell, Ethernet-Spezifikationen, TCP/IP • Aufbau einfacher HTML-Seiten (statische Webseiten) • Formulare in HTML, Übermittlung durch Mail-Programme <p>Lokale Datenbanken</p> <ul style="list-style-type: none"> • Grundbegriffe und Zweck der Datenbanken • Anwenderansichten auf Datenbanken • Modellierungen per ERM • Relationale Datenbanken • Datenbankanfragen mit SQL-Abfragen <p>Web-Datenbanken</p> <ul style="list-style-type: none"> • Webserver (xAMPP) • myPHPAdmin • PHP-Skripte zum Zugriff auf Datenbanken • Datamining und Datenschutz 	<ul style="list-style-type: none"> • „Warriors of the Net“ und dessen Analyse • Netzwerk-Grundlagen im Haushalt und in der Schule • Software in Netzwerken (Browser, FTP-Transfer, WireShark, Terminal) • HTML-Crashkurs: Tags, schließende Tags zur Webseitengestaltung • Web-Administratoren Tools: FTP, SSH, HTML-Editoren • Handouts vgl. Moodle-Kurs (selbstentwickelte Zusammenfassungen nach: A. Kemper, A. Eickler: Datenbanksysteme. Eine Einführung, München 2004) 	<p>3.6.4</p> <p>3.6.1 (Datenbanken)</p> <p>3.6.3 (Programmiersprachen)</p>