



Schulinternes Curriculum

Informatik

nach dem

**Kernlehrplan für die Sekundarstufe I
Gymnasium
in NRW**

Juni 2022

Erprobungsstufe 6

G9

Mittelstufe 8, 9 und 10

G9

Oberstufe EF, Q1 und Q2

Inhaltsverzeichnis

I.	Sekundarstufe I.....	4
1.	Jahrgangstufe 6	5
2.	Jahrgangstufe 8 bzw. 9	15
3.	Jahrgangsstufe 9 bzw. 10	16
II.	Sekundarstufe II.....	16
1.	Übersicht über die Unterrichtsvorhaben	18
a.	Die Einführungsphase der Oberstufe (EF)	18
b.	Die Qualifikationsphase (Q1 und Q2) - Grundkurs.....	28
c.	Die Qualifikationsphase (Q1 und Q2) - Leistungskurs.....	53
III.	Grundsätze der Leistungsbewertung im Fach Informatik	55
IV.	Formen der Leistungsbewertung	56
1.	Beurteilungsbereich schriftliche Lernerfolgsüberprüfungen	56
a.	Klausuren.....	56
2.	Beurteilungsbereich sonstige Mitarbeit.....	57
a.	Kriterien und Indikatorenkatalog zur Bewertung der sonstigen Mitarbeit.....	57
3.	Verhältnis schriftliche Leistungsüberprüfungen zu sonstiger Mitarbeit / Gesamtnote	58
V.	Anhang I: Kriterien- und Indikatorenkatalog zur Leistungsbewertung	59

I. Sekundarstufe I

Das Fach Informatik ist am Städtischen Gymnasium Leichlingen ein Wahlfach und wird erstmalig in der Jahrgangsstufe 6 und im Rahmen des Wahlpflichtbereichs II in der Jahrgangsstufe 9 angeboten. Darüber hinaus wird schulweit allen Schülerinnen und Schülern angeboten an den jährlich stattfindenden Wettbewerben Informatik-Biber und Jugendwettbewerb Informatik teilzunehmen. Die unterrichtlichen Inhalte der einzelnen Jahrgangsstufen verteilen sich wie folgt:

1. Jahrgangstufe 6

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.1 Informationen und Daten</p> <p><i>Was sind Daten und wie werden sie zu Informationen?</i></p> <p>ca. 4 Ustd.</p>	<p>Information und Daten</p> <ul style="list-style-type: none"> • Daten und ihre Codierung • Informationsgehalt von Daten 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • formulieren Fragen zu einfachen informatischen Sachverhalten (A), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • beschreiben einfach informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • erläutern den Datenbegriff anhand von Beispielen aus ihrer Erfahrungswelt (A), • erläutern den Zusammenhang und die Bedeutung von Information und Daten (A), • stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI), • interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI), • erläutern Einheiten von Datenmengen (A / KK), • vergleichen Datenmengen hinsichtlich ihrer Größe mithilfe anschaulicher Beispiele aus ihrer Lebenswelt (DI).
<p>Vorhabenbezogene Absprachen und Empfehlungen</p> <p>Hinweise: Die SuS sollen die verschiedenen Einheiten von Datenmengen kennenlernen und passende Speichermedien für Dateien auswählen.</p>		

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.2 Nachrichten in einen Code umwandeln</p> <p><i>Wie kann man mit einem Computer kommunizieren?</i></p>	<p>Information und Daten</p> <ul style="list-style-type: none"> • Daten und ihre Codierung • Informationsgehalt von Daten 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • formulieren Fragen zu einfachen informatischen Sachverhalten (A), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • interpretieren informatische Darstellungen (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fach-

ca. 6 Ustd.		begriffen sachgerecht (KK), <ul style="list-style-type: none"> kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme (KK). <i>Konkretisierte Kompetenzerwartungen</i> <ul style="list-style-type: none"> stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI), nennen Beispiele für die Codierung von Daten aus ihrer Erfahrungswelt (DI), codieren und decodieren Daten unter Verwendung des Binärsystems (MI), interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI).
<p>Vorhabenbezogene Absprachen und Empfehlungen</p> <p>... zur Vernetzung: Informationen aus Daten werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Bereich der Kryptologie, der künstlichen Intelligenz oder dem Datenbewusstsein.</p> <p>... zu Synergien: Mathematik (Stellenwertsystem).</p> <p>Hinweise: z. B. wird das Winkeralphabet und der Morsecode zur Übermittlung von Nachrichten besprochen. Das Codieren und Decodieren von Daten erfolgt u. a. unter Verwendung des Binärsystems.</p>		

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
6.3 Kryptologie <i>Wie kann man Nachrichten geheim übermitteln?</i> ca. 6 Ustd.	Information und Daten <ul style="list-style-type: none"> Informationsgehalt von Daten Verschlüsselungsverfahren 	<i>Übergeordnete Kompetenzerwartungen</i> <ul style="list-style-type: none"> formulieren Fragen zu einfachen informatischen Sachverhalten (A), äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen (A), beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), interpretieren informatische Darstellungen (DI), beschreiben einfach informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK), Kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbei-

		<p>tung einfacher informatischer Probleme (KK).</p> <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • stellen eine ausgewählte Information in geeigneter Form als Daten formal-sprachlich oder graphisch dar (DI), • interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI), • erläutern ein einfaches Transpositionsverfahren als Möglichkeit der Verschlüsselung (DI), • vergleichen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (DI).
<p>Vorhabenbezogene Absprachen und Empfehlungen</p> <p>... zur Vernetzung: Informationen aus Daten zu erhalten und diese zu entschlüsseln, spielt im Unterrichtsverlauf in Informatik immer wieder eine Rolle, z. B. Information und Daten.</p> <p>Hinweise: Als Transpositionsverfahren bieten sich je nach Lerngruppe unterschiedliche an, z. B. die Skytale oder die Verschlüsselung mit Schablonen. Ergänzend können noch verschiedenen Substitutionsverfahren, wie beispielsweise die Cäsarverschlüsselung, besprochen werden. Im Anschluss können als Vertiefung polyalphabetische Verschlüsselungsverfahren (wie die Vigenère-Verschlüsselung) erarbeitet werden.</p>		

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.4 Umgang mit Informatiksystemen</p> <p><i>Woraus besteht ein Computer und wie verarbeitet er Daten?</i></p> <p>ca. 4 Ustd.</p>	<p>Informatiksysteme</p> <ul style="list-style-type: none"> • Aufbau und Funktionsweise von Informatiksystemen • Anwendung von Informatiksystemen 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen (A), • erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen (A), • begründen die Auswahl eines Informatiksystems (A), • erstellen informatische Modelle zu gegebenen Sachverhalten (MI), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • benennen Beispiele für (vernetzte) Informatiksysteme aus ihrer Erfahrungswelt

		<p>(DI),</p> <ul style="list-style-type: none"> • benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI), • beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung (DI), • vergleichen Möglichkeiten der Datenverwaltung hinsichtlich ihrer spezifischen Charakteristika (u. a. Speicherort, Kapazität, Aspekte der Datensicherheit) (A), • setzen zielgerichtet Informatiksysteme zur Verarbeitung von Daten ein (MI), • erläutern Prinzipien der strukturierten Dateiverwaltung (A), • setzen Informatiksysteme zur Kommunikation und Kooperation ein (KK).
<p>Vorhabenbezogene Absprachen und Empfehlungen</p> <p>... zur Vernetzung: Informatiksysteme werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Bereich Algorithmen oder Automaten.</p> <p>... zu Synergien: Strukturierte Bezeichnung und Ablage von Dateien beispielsweise in der Informatischen Grundbildung.</p> <p>Hinweise: Die SuS lernen insbesondere, wie man Dateien strukturiert verwaltet und benennt, da diese Kompetenz im weiterem Schul- / Arbeitsleben benötigt wird.</p>		

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.5 Von der Anweisung zum Algorithmus</p> <p><i>Wo begegnen uns und wie helfen uns Algorithmen im Alltag?</i></p> <p>ca. 8 Ustd.</p>	<p>Algorithmen I</p> <ul style="list-style-type: none"> • Algorithmen und algorithmische Grundkonzepte 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • erstellen informatische Modelle zu gegebenen Sachverhalten (MI), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • stellen informatische Sachverhalte in geeigneter Form dar (DI), • interpretieren informatische Darstellungen (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK), • strukturieren gemeinsam eine Lösung für ein informatisches Problem (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p>

		<ul style="list-style-type: none"> • formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI), • überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI), • führen Handlungsvorschriften schrittweise aus (MI), • identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI), • ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI), • bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A).
--	--	--

Vorhabenbezogene Absprachen und Empfehlungen

... zur Vernetzung:

Algorithmen werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Bereich Informatiksysteme oder Automaten.

... zu Synergien:

Mathematik (Systematisierung von Rechenoperationen, z. B. schriftliches Dividieren)

Hinweise:

Als Algorithmus im Alltag kann, z.B. der Weg zur Schule thematisiert werden. Außerdem können die Kinder durch einfache Zeichenanleitungen bereits das Programmieren in der textbasierten Programmiersprache (XLogo) vorbereiten.

Auch die Veranschaulichung von Algorithmen in Programmablaufplänen findet hier bereits statt.

Die Implementation von Algorithmen erfolgt später in einer textbasierten Programmiersprache und einer visuellen Programmiersprache.

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.6 Textbasiertes Programmieren</p> <p><i>Wie steuert man eine Schildkröte über den Bildschirm?</i></p> <p>ca. 6 Ustd.</p>	<p>Algorithmen II</p> <ul style="list-style-type: none"> • Algorithmen und algorithmische Grundkonzepte • Implementation von Algorithmen 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • bewerten das Ergebnis einer informatischen Modellierung (A), • erstellen informatische Modelle zu gegebenen Sachverhalten (MI), • implementieren informatische Modelle unter Verwendung algorithmischer Grundstrukturen (MI), • überprüfen Modelle und Implementierungen (MI), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • stellen informatische Sachverhalte in geeigneter Form dar (DI),

		<ul style="list-style-type: none"> • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK), • kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme (KK), • strukturieren gemeinsam eine Lösung für ein informatisches Problem (KK), • dokumentieren gemeinsam ihren Arbeitsprozess und ihre Ergebnisse auch mithilfe digitaler Werkzeuge (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • führen Handlungsvorschriften schrittweise aus (MI), • identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz (MI), • implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI), • überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI), • ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI), • bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A).
--	--	---

Vorhabenbezogene Absprachen und Empfehlungen

... zur Vernetzung:

Algorithmen werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Bereich Informatiksysteme oder Automaten.

Hinweise:

Im Anschluss an die Zeichenanleitungen bietet sich das Programmieren z. B. in der textbasierten Programmiersprache XLogo an um zunächst einfache algorithmische Grundstrukturen (wie Sequenzen) zu implementieren. Je nach Lerngruppe können jedoch auch schon Verzweigungen und Schleifen behandelt werden. Diese werden ansonsten später in einer visuellen Programmierumgebung implementiert.

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
6.7 Automaten in unserer Lebenswelt	Automaten und künstliche Intelligenz <ul style="list-style-type: none"> • Aufbau und Wirkungsweise ein- 	<i>Übergeordnete Kompetenzerwartungen</i> <ul style="list-style-type: none"> • formulieren Fragen zu einfachen informatischen Sachverhalten (A),

<p><i>Was macht einen Automaten aus?</i></p> <p>ca. 4 Ustd.</p>	<p>facher Automaten</p>	<ul style="list-style-type: none"> • äußern Vermutungen zu informatischen Sachverhalten auf Basis von Alltagsvorstellungen oder Vorwissen (A), • erstellen informatische Modelle zu gegebenen Sachverhalten (MI), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • erläutern die Funktionsweise eines Automaten aus ihrer Lebenswelt (A), • stellen Abläufe in Automaten graphisch dar (DI).
<p>Vorhabenbezogene Absprachen und Empfehlungen</p> <p>... zur Vernetzung: Der Aufbau und die Wirkungsweise von Automaten wird im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Bereich der künstlichen Intelligenz.</p> <p>Hinweise: Als Einstiegsbeispiel kann das strukturierte Vorgehen bei der Bedienung eines Parkautomaten / Getränkeautomaten betrachtet werden. In dem Zusammenhang kann man auf die verschiedenen Zustände eines Automaten eingehen.</p>		

<p>Unterrichtsvorhaben</p>	<p>Inhaltsfelder Inhaltliche Schwerpunkte</p>	<p>Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler</p>
<p>6.8 Künstliche Intelligenz</p> <p><i>Wie lernt eine Maschine?</i></p> <p>ca. 8 Ustd.</p>	<p>Automaten und künstliche Intelligenz</p> <ul style="list-style-type: none"> • Maschinelles Lernen mit Entscheidungsbäumen • Maschinelles Lernen mit neuronalen Netzen 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • formulieren Fragen zu einfachen informatischen Sachverhalten (A), • äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen (A), • erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen (A), • erstellen informatische Modelle zu gegebenen Sachverhalten (MI), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • stellen informatische Sachverhalte in geeigneter Form dar (DI), • interpretieren informatische Darstellungen (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK).

		<p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • benennen Anwendungsbeispiele künstlicher Intelligenz aus ihrer Lebenswelt (A), • stellen das Grundprinzip eines Entscheidungsbaumes enaktiv als ein Prinzip des maschinellen Lernens dar (DI), • beschreiben die grundlegende Funktionsweise künstlicher neuronaler Netze in verschiedenen Anwendungsbeispielen (KK).
--	--	---

Vorhabenbezogene Absprachen und Empfehlungen

Hinweise:

Entscheidungsbäume können anhand der Tierpflegergeschichte anschaulich näher gebracht werden. Mithilfe von „Google-QuickDraw“ sehen die SuS ein Beispiel für Deep-Learning. Das bestärkende Lernen kann durch das Spiel „Schlag das Krokodil“ veranschaulicht werden.

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.9 Programmieren in einer visuellen Programmiersprache</p> <p><i>Kleine Spiele für den Computer entwickeln</i></p> <p>ca. 8 Ustd.</p>	<p>Algorithmen III</p> <ul style="list-style-type: none"> • Algorithmen und algorithmische Grundkonzepte • Implementation von Algorithmen 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • bewerten das Ergebnis einer informatischen Modellierung (A), • erstellen informatische Modelle zu gegebenen Sachverhalten (MI), • implementieren informatische Modelle unter Verwendung algorithmischer Grundstrukturen (MI), • überprüfen Modelle und Implementierungen (MI), • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • stellen informatische Sachverhalte in geeigneter Form dar (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK), • kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme (KK), • strukturieren gemeinsam eine Lösung für ein informatisches Problem (KK). • dokumentieren gemeinsam ihren Arbeitsprozess und ihre Ergebnisse auch mithilfe digitaler Werkzeuge (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder

		<p>ein Struktogramm (MI),</p> <ul style="list-style-type: none"> • führen Handlungsvorschriften schrittweise aus (MI), • identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI), • implementieren Algorithmen in einer visuellen Programmiersprache (MI), • implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI), • überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI), • ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI), • bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A).
--	--	--

Vorhabenbezogene Absprachen und Empfehlungen

... zur Vernetzung:

Algorithmen werden im weiteren Unterrichtsverlauf in Informatik immer wieder eine Rolle spielen, z. B. im Bereich Informatiksysteme oder Automaten.

... zu Synergien:

Mögliche Kooperation mit dem Fach Physik über die Funktionen von Sensoren.

Hinweise:

Als visuelle Programmiersprache bietet sich Scratch oder NEPO an. Die SuS können selber den Calliope programmieren.

Als algorithmische Grundstrukturen kommen nun die Schleifen und Verzweigungen hinzu. Auch Parameter und Variablen werden eingesetzt. Dadurch können komplexere Programme implementiert werden.

Unterrichtsvorhaben	Inhaltsfelder Inhaltliche Schwerpunkte	Schwerpunkte der Kompetenzentwicklung Die Schülerinnen und Schüler
<p>6.10 Datenbewusstsein</p> <p><i>Welche Informationen kann man aus meinen Daten über mich ableiten?</i></p>	<p>Informatik, Mensch und Gesellschaft</p> <ul style="list-style-type: none"> • Informatiksysteme in der Lebens- und Arbeitswelt • Datenbewusstsein • Datensicherheit und Sicherheitsregeln 	<p><i>Übergeordnete Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • formulieren Fragen zu einfachen informatischen Sachverhalten (A), • äußern Vermutungen zu informatischen Sachverhalten auf Basis von Alltagsvorstellungen oder Vorwissen (A), • erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen (A), • erstellen informatische Modelle zu gegebenen Sachverhalten (MI),

ca. 6 Ustd.		<ul style="list-style-type: none"> • beschreiben einfache Darstellungen von informatischen Sachverhalten (DI), • stellen informatische Sachverhalte in geeigneter Form dar (DI), • interpretieren informatische Darstellungen (DI), • beschreiben einfache informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht (KK), • kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme (KK). <p><i>Konkretisierte Kompetenzerwartungen</i></p> <ul style="list-style-type: none"> • beschreiben an Beispielen die Bedeutung von Informatiksystemen in der Lebens- und Arbeitswelt (KK), • benennen an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen auf ihre Lebens- und Erfahrungswelt (A/KK), • anstelle der vorherigen KE: erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK), • beschreiben anhand von ausgewählten Beispielen die Verarbeitung und Nutzung personenbezogener Daten (DI), • erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A), • beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A).
<p>Vorhabenbezogene Absprachen und Empfehlungen</p> <p>... zur Vernetzung: Der Umgang mit Daten und die damit verbundenen Regeln spielen im Unterrichtsverlauf in Informatik immer wieder eine Rolle.</p> <p>... zu Synergien: Soziales Lernen</p> <p>Hinweise: Die SuS sollen sensibilisiert werden, welche Daten sie im Internet hinterlassen (digitaler Fingerabdruck) und welche Daten große Konzerne zu welchem Zweck von ihnen speichern.</p>		

2. Jahrgangstufe 8 bzw. 9

Im ersten Halbjahr werden Tabellenkalkulations- und Textverarbeitungsprogramme am Beispiel Microsoft Office mit folgenden Schwerpunkten behandelt:

- Allgemeine Einführung in Tabellenkalkulationen
- Relative und absolute Zellbezüge
- WENN- und SVERWEIS-Formel
- Bedingte Formatierungen
- Erstellen und Verändern von verschiedenen Diagrammtypen zur Visualisierung von Daten
- Verknüpfung von Tabellenkalkulation und Textverarbeitung anhand von Serienbriefen

Die Schülerinnen und Schüler verarbeiten Daten mit Hilfe von Informatiksystemen und verwenden hierbei arithmetische und logische Operationen. Die Aufgabe in dieser Phase berücksichtigen in besonderem Maße die soziale Interaktion und damit das gemeinsame Lernen. Somit ergibt sich eine natürliche Möglichkeit der Binnendifferenzierung nach dem Prinzip „Lernen durch Lehren“.

Im zweiten Halbjahr wird zum ersten Mal eine Programmiersprache thematisiert – Scratch.

- Einführung in die Möglichkeiten, die Arbeitsweise und Oberfläche von Scratch
- Einführung in die algorithmische Denkweise anhand einfacher Kontrollstrukturen
- Übersicht über verschiedene Schleifenkonstrukte
- Einführung der Begriffe Objekt, Eigenschaften und Verhalten
- Einführung des Modellierungskonzepts
- Implementierung kleiner Spiele und Simulationen

Die Schülerinnen und Schüler entwerfen Algorithmen unter Verwendung des Variablenkonzepts und von Kontrollstrukturen. Sie implementieren und testen diese Algorithmen anschließend in einer Programmierumgebung. Zum Ende des Halbjahres wählen die Schülerinnen und Schüler ein ihnen Neigungen entsprechendes Thema, in dem sie die erworbenen Kompetenzen im Rahmen eines Projekts anwenden können und dies abschließend in der Kursgemeinschaft präsentieren.

3. Jahrgangsstufe 9 bzw. 10

Die Stufe 9 bzw. 10 steht ganz im Lichte von Internet, WWW und HTML/CSS. Im ersten Halbjahr werden insbesondere folgende Inhalte thematisiert:

- Grundsätzliche Funktionsweise und darauf aufbauenden Internetdienste (WWW, Email, P2P, etc.)
- Gefahren und Schutzmechanismen der einzelnen Internetdienste
- Unterschiedliche Grafikformate und deren Anwendungsgebiete
- Einführung in HTML
 - Grundsätzlicher Aufbau eines HTML-Dokuments
 - Einfügen von Bildern und Links anhand relativer und absoluter Pfadangaben
 - Tabellen zur Strukturierung der eigenen Inhalte
- Einführung in CSS
 - Einführung in die Möglichkeiten und Grenzen von CSS
 - Einbinden einer Datei mit CSS-Befehlen
 - Erstellung und Anwendung von (Pseudo-)Klassen

Die Schülerinnen und Schüler beschreiben die Gefährdung von Daten durch Defekte und Schadsoftware und benennen Maßnahmen zum Schutz von Daten und zur sicheren Kommunikation in Netzwerken. Des Weiteren erstellen sie syntaktisch korrekte Quelltexte in einer Dokumentenbeschreibungssprache. Bei der Erstellung einer eigenen Homepage können die Schülerinnen und Schüler ihre Kreativität einbringen und sie nach ihren individuellen Wünschen gestalten.

Im Sinne des Spiralcurriculums werden im zweiten Halbjahr einerseits bekannte Inhalte aus 8.2 wiederholt und andererseits auf eine neue Programmiersprache übertragen – JavaScript.

- Allgemeiner Aufbau einer Datei mit JavaScript
- Einführung des Variablenkonzepts, Schleifen und weitere Kontrollstrukturen in JavaScript
- Modellierung und Implementierung kleiner Programme

Die Schülerinnen und Schüler entwickeln mit Hilfe ihnen bekannter Befehle selbstständig Lösungen für komplexere Problemstellungen, wobei sie ihre Ergebnisse interpretieren und die erstellten Quelltexte u.a. auf syntaktische Korrektheit analysieren.

II. Sekundarstufe II

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

In der „Übersicht über die Unterrichtsvorhaben“ wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Die Übersicht dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompeten-

zen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen (z.B. die Teilnahme an Wettbewerben), Bearbeitung aktueller Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Die Beispiele und Materialien in den konkretisierten Unterrichtsvorhaben sollen empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

1. Übersicht über die Unterrichtsvorhaben

a. Die Einführungsphase der Oberstufe (EF)

<p><u>Unterrichtsvorhaben EF-I</u></p> <p><i>Grundlagen der Modellierung und Implementierung anhand eines einfachen Programmierbeispiels</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Modellieren• Implementieren• Darstellen und Interpretieren• Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 10 Stunden</p>	<p><u>Unterrichtsvorhaben EF-II</u></p> <p><i>Wenn du häufig denselben oder sehr ähnlichen Programmcode schreibst, gibt es wahrscheinlich eine elegantere Lösung -Grundlagen der Übergabe von Parametern und Verwendung von Rückgaben anhand einfacher Funktionen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Implementieren• Kommunizieren und Kooperieren• Modellieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">• Daten und ihre Strukturierung• Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">• Objekte und Klassen• Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 15 Stunden</p>
<p><u>Unterrichtsvorhaben EF-III</u></p> <p><i>Die erste eigene Klasse - Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Darstellen und Interpretieren• Implementieren• Kommunizieren und Kooperieren• Modellieren	<p><u>Unterrichtsvorhaben EF-IV</u></p> <p><i>Such- und Sortieralgorithmen anhand kontextbezogener Beispiele</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">• Argumentieren• Modellieren• Darstellen und Interpretieren

<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Digitalisierung <p>Zeitbedarf: 20 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Daten und ihre Strukturierung <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Algorithmen zum Suchen und Sortieren • Analyse, Entwurf und Implementierung einfacher Algorithmen <p>Zeitbedarf: 15 Stunden</p>
<p><u>Unterrichtsvorhaben EF-V</u></p> <p><i>Wenn du häufig denselben oder sehr ähnlichen Programmcode schreibst, gibt es wahrscheinlich eine elegantere Lösung II - Erweiterte Modellierung und Implementierung von Klassen- und Objektbeziehungen durch das Prinzip der Vererbung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Darstellen und Interpretieren • Kommunizieren und Kooperieren • Modellieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen <p>Zeitbedarf: 15 Stunden</p>	<p><u>Unterrichtsvorhaben EF-VI</u></p> <p><i>Geschichte, Auswirkung und Zukunft der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatik, Mensch und Gesellschaft • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Wirkungen der Automatisierung • Geschichte der automatischen Datenverarbeitung • Digitalisierung <p>Zeitbedarf: 12 Stunden</p>

Unterrichtsvorhaben EF-I

Thema: Grundlagen der Modellierung und Implementierung anhand einfacher Objekte

Leitfrage: Wie lassen sich Abläufe einer Simulation informatisch beeinflussen?

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Programmierung ein. Dazu werden zunächst einfache geometrische Formen auf dem Bildschirm dargestellt und diese anschließend variiert. Im Anschluss wird mit der Realisierung erster Projekte begonnen. Am Beispiel eines vorgegebenen Programms lernen die SuS Objekte und Variablen kennen. Durch Variation des Beispiels entdecken sie selbständig die Bedeutung einzelner Parameter.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Kennenlernen graphischer Parameter (a) Figuren zeichnen (b) Ball zeichnen	Die Schülerinnen und Schüler <ul style="list-style-type: none"> entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), testen Programme schrittweise anhand von Beispielen (I), analysieren und erläutern einfache Algorithmen und Programme (A), modifizieren einfache Algorithmen und Programme (I), implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), interpretieren Fehlermeldungen und korrigieren den Quellcode (I). 	<i>Materialien:</i> ErsteSchritte.java
2. Bewegungsanimationen am Beispiel einfacher grafischer Objekte (a) Kontinuierliche Verschiebung eines Objekts mit Hilfe einer Schleife (b) Überprüfung einer Bedingung für eine Animationsschleife (IF-Anweisungen) (c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen	(siehe oben)	<i>Beispiel:</i> Figuren zeichnen Schülerinnen und Schüler zeichnen mit der Funktion drawRect verschiedene Figuren auf den Bildschirm <i>Beispiel:</i> Haus vom Nikolaus Die Schülerinnen und Schüler zeichnen das Haus vom Nikolaus – Methode: In Dokumentation nach passenden Funktionen suchen, z.B. drawLine <i>Beispiel:</i> Rollen und Abprallen eines Balles Die Schülerinnen und Schüler simulieren das Rollen und Abprallen eines Balles an gesetzten Grenzen.

Unterrichtsvorhaben EF-II

Thema: Wenn du häufig denselben oder sehr ähnlichen Programmcode schreibst, gibt es wahrscheinlich eine elegantere Lösung -Grundlagen der Übergabe von Parametern und Verwendung von Rückgaben anhand einfacher Funktionen

Leitfragen: *Wie kann die Häufung von gleichen Befehlen vermieden werden? Wie kann ein Programm übersichtlicher strukturiert werden? Wie kann einer Funktion eine zusätzliche Information (Parameter) übergeben werden?*

Vorhabenbezogene Konkretisierung:

Dem zweiten Unterrichtsvorhaben liegt die Häufung von gleichen Befehlen oder Befehlsblöcken zu Grunde. Den Schülerinnen und Schülern soll anhand einer einfachen Problemstellung heraus auffallen, dass simples Wiederholen (z.B. durch Kopieren und Einfügen) von Programmcode Zeit und Übersicht verloren geht und, durch die Auslagerung dieses Programmcodes in Funktionen, Zurückgewonnen werden kann.

Darauf aufbauend werden Parameter als „zusätzliche Information“ für die eigenen und fremden Funktionen eingeführt, welche notwendig sind, um dieselbe Funktion für leicht unterschiedliche Zwecke verwenden zu können.

Abschließend werden die bisherigen Funktionen (Aufträge) um die Möglichkeit einer Antwort/eines Rückgabewertes erweitert (Anfragen).

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Die erste eigene Funktion (a) Problematisierung durch Häufungen von Befehlen im Programmcode (b) Einführung der Begriffe Funktion und Signatur	Die Schülerinnen und Schüler <ul style="list-style-type: none"> ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I). 	<i>Beispiel:</i> Zeichnung eines Objekts (z.B. Haus-vom-Nikolaus) mittels der Klasse Graphics und anschließender Auslagerung in eine eigene Funktion
2. Parameter als zusätzliche Information (a) Erweiterung der eigenen Funktion um einen oder mehrere Parameter		<i>Beispiel:</i> Erweiterung der eigenen Funktion um Parameter für die Ortsbestimmung des Hauses.
3. Rückgaben von Funktionen verwenden (a) Einführung des Begriffs des Rückgabetyps		<i>Beispiel:</i> Berechnung des umbauten Raumes der gezeichneten Häuser

Unterrichtsvorhaben EF-III

Thema: Die erste eigene Klasse - Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen

Leitfragen: Was ist ein Objekt? Was kann ein Objekt? Wie wird ein Objekt erstellt?

Vorhabenbezogene Konkretisierung:

Im Zentrum des dritten Unterrichtsvorhabens stehen die Begriffe Klasse und Objekt. Anhand eines Spiels, einer Simulation oder realer Gegenstände werden zuerst die Bestandteile eines Objekts – Name, Attribut und Methoden – eingeführt und anschließend durch das Konzept der Klasse als Bauplan für Objekte erweitert. Abschließend wird, ausgehend von der Tastatursteuerung eines Spiels, die ASCII-Tabelle erläutert und verwendet.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Das Objekt -> Die Klasse</p> <p>(a) Erarbeitung des Aufbaus eines Objekts</p> <p>(b) Einführung des Aufbaus einer Klasse</p> <p>(c) Die Rolle des Konstruktors und Implementierung einer zuvor modellierten Klasse</p> <p>(d) Selbstständige Erweiterung des Modells/der Klasse</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), stellen den Zustand eines Objekts dar (D), modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), 	<p><i>Beispiel:</i> Tische im Klassenraum, Tiere, Autos als reale Objekte, danach der Ball aus UV EF-I als eigenes Objekt.</p> <p><i>Beispiel:</i> Mehrere Objekte vom Typ Ball aus UV EF-I implementieren</p> <p><i>Beispiel:</i> Den Ball (rudimentär) mit der Tastatur steuern können</p>
<p>2. Modellierung der zweiten eigenen Klasse</p> <p>(a) Einführung der UML-Diagramme anhand der bestehenden Klassen</p> <p>(b) Erweiterung dieses UML-Diagramms um eine weitere Klasse</p> <p>(c) Analyse unterschiedlicher Varianten dieser Modellierung</p> <p>(d) Implementierung der eigenen Klasse</p>	<ul style="list-style-type: none"> ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), stellen die Kommunikation zwischen Objekten grafisch dar (M), 	<p><i>Beispiel:</i> Erweiterung des Programms zu einer Variante des Spiels Pong.</p>

<p>3. Zahlen und Zeichen im Binärcode</p> <p>(a) Anhand eines geeigneten Beispiels die Problematik der Speicherung von Informationen als Binärcode verdeutlichen</p> <p>(b) Umrechnung von Dezimal- in Dualzahlen</p> <p>(c) Einführung der ASCII-Tabelle</p>	<ul style="list-style-type: none"> • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D), • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen ganze Zahlen und Zeichen in Binär-codes dar (D), • interpretieren Binär-codes als Zahlen und Zeichen (D). 	<p><i>Beispiel:</i> Der Datentyp int. Die Tastatursteuerung im Spiel Pong.</p>
--	---	--

Unterrichtsvorhaben EF-IV

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Leitfragen: *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

Vorhabenbezogene Konkretisierung:

Zunächst werden Objekte aus der Lebenswelt der Schülerinnen und Schüler (z. B. Spielkarten, nummerierte USB-Sticks) sortiert. Dabei werden Strategien zur Sortierung von den Schülerinnen und Schülern selbst erarbeitet und hinsichtlich der Anzahl notwendiger Vergleiche auf ihre Effizienz untersucht.

Bevor einzelne Sortieralgorithmen implementiert werden, wird anhand eines Beispiels verdeutlicht, dass es nützlich ist, wenn viele gleichartige Felder verarbeitet werden sollen, diese als Array zu definieren.

Schließlich werden einzelne erarbeitete Strategien unter Verwendung von Arrays implementiert. Die Schülerinnen und Schüler sollen auf diese Weise das *Sortieren* mit mindestens einem Sortieralgorithmus kennen lernen.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Explorative Erarbeitung eines Sortierverfahrens</p> <p>(a) Sortierprobleme im Alltagskontext (z.B. Telefonbuch, Bundesligatabelle, Spielkarten usw.)</p> <p>(b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus</p> <p>(c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler</p> <p>2. Erstellen und Verwalten größerer Mengen einfacher Objekte</p> <p>(a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife)</p> <p>(b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays)</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • entwerfen einen weiteren Algorithmus zum Sortieren (M), • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A). 	<p><i>Beispiel:</i> Sortieren mit Waage Die Schülerinnen und Schüler bekommen die Aufgabe, kleine, optisch identische Kunststoffbehälter aufsteigend nach ihrem Gewicht zu sortieren. Dazu steht ihnen eine Balkenwaage zur Verfügung, mit deren Hilfe sie das Gewicht zweier Behälter vergleichen können.</p>
<p>2. Implementieren eines Algorithmus und Effizienzbetrachtungen</p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Implementierung zweier einfacher Sortieralgorithmen</p> <p>(c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche</p> <p>(d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs</p>		<p><i>Beispiele:</i> Sortieren durch Auswählen, Sortieren durch Einfügen</p>

Unterrichtsvorhaben EF-V

Thema: Wenn du häufig denselben oder sehr ähnlichen Programmcode schreibst, gibt es wahrscheinlich eine elegantere Lösung III - Erweiterte Modellierung und Implementierung von Klassen- und Objektbeziehungen durch das Prinzip der Vererbung

Leitfragen: *Wie lassen sich Redundanzen bei der Modellierung von Klassen vermeiden?*

Vorhabenbezogene Konkretisierung:

In Spielen gibt es viele Klassen, die viele Gemeinsamkeiten haben – X-Koordinate, Y-Koordinate, Bild, Bewegung, entsprechende getter- und setter-Methoden. Zu Beginn dieses Unterrichtsvorhabens sollten die Schülerinnen und Schüler in der Lage sein, diese Redundanz zu erkennen und am Ende, sie zu mittels Vererbung zu umgehen.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Ein Auto und Fahrrad haben Gemeinsamkeiten</p> <ul style="list-style-type: none"> (a) Erarbeitung des konkreten Ausgangsproblems (b) Einführung des Prinzips der Vererbung (c) Vergleich der Modellierung des Problems mit und ohne Vererbung (d) Implementierung der Ober- und Unterklasse 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modellieren Klassen unter Verwendung von Vererbung (M), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M). 	<p><i>Beispiel:</i> Gemeinsamkeiten in einem Spiel mit ca. 5 Klassen erkennen und mittels Vererbung beseitigen</p>
<p>2. Sichtbarkeitsbereiche</p> <ul style="list-style-type: none"> (a) Wiederholung/Konkretisierung der Sichtbarkeitsmodifikatoren public und private (b) Erweiterung um den Sichtbarkeitsmodifikator protected 		

Unterrichtsvorhaben EF-VI

Thema: Geschichte, Auswirkung und Zukunft der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Leitfrage: Welche Entwicklung durchlief die moderne Datenverarbeitung, welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus und wie sieht die Zukunft der Digitalen DV aus?

Vorhabenbezogene Konkretisierung:

Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Schülerinnen und Schüler sollen selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten. Diese Themenbereiche werden in Kleingruppen bearbeitet und in Form von Plakatpräsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler</p> <p>(a) Mögliche Themen zur Erarbeitung in Kleingruppen:</p> <ul style="list-style-type: none">„Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“„Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“„Von Neumann-Architektur“	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K),nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),	<p><i>Beispiel:</i> Ausstellung oder Referate zu informatischen Themen</p> <p><i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, im Selbstlernzentrum, in öffentlichen Bibliotheken, usw.</p>

<ul style="list-style-type: none"> • „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ • „Das EVA-Prinzip“ • „Was ist Java?“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ <p>(b) Vorstellung und Diskussion durch Schülerinnen und Schüler</p>	<ul style="list-style-type: none"> • beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A). 	
<p>2. Vertiefung des Themas Datenschutz</p> <p>(a) Erarbeitung grundlegender Begriffe des Datenschutzes</p> <p>(b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler</p> <p>(c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“</p>		<p><i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung.</p> <p><i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz</p>

b. Die Qualifikationsphase (Q1 und Q2) - Grundkurs

Die Qualifikationsphase baut entscheidend auf den Inhalten der Einführungsphase auf und erweitert insbesondere die Fähigkeiten und Kenntnisse im Bereich des objektorientierten Modellierens und des Implementierens von kontextbezogenen Anwendungen.

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><i>Wiederholung der objektorientierten Modellierung und Programmierung anhand einer kontextbezogenen Problemstellung</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><i>Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p><i>Suchen und Sortieren auf linearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren 	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p><i>Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren

<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 16 Stunden</p>	<p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Datenbanken • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache • Sicherheit <p>Zeitbedarf: 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q1-V</u></p> <p><i>Sicherheit und Datenschutz in Netzstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: 15 Stunden</p>	

Unterrichtsvorhaben Q1-I

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Leitfragen: *Wie modelliert und implementiert man zu einer Problemstellung in einem geeigneten Anwendungskontext Java-Klassen inklusive ihrer Attribute, Methoden und Beziehungen? Wie kann man die Modellierung und die Funktionsweise der Anwendung grafisch darstellen?*

Vorhabenbezogene Konkretisierung:

Zu einer Problemstellung in einem Anwendungskontext soll eine Java-Anwendung entwickelt werden. Die Problemstellung soll so gewählt sein, dass für diese Anwendung die Verwendung einer abstrakten Oberklasse als Generalisierung verschiedener Unterklassen sinnvoll erscheint und eine Klasse durch eine Unterklasse spezialisiert werden kann. Um die Aufgabe einzugrenzen, können (nach der ersten Problemanalyse) einige Teile (Modellierungen oder Teile von Java-Klassen) vorgegeben werden.

Die Schülerinnen und Schülern erläutern und modifizieren den ersten Entwurf und modellieren sowie implementieren weitere Klassen und Methoden für eine entsprechende Anwendung. Klassen und ihre Beziehungen werden in einem Implementationsdiagramm dargestellt. Dabei werden Sichtbarkeitsbereiche zugeordnet. Exemplarisch wird eine Klasse dokumentiert. Der Nachrichtenaustausch zwischen verschiedenen Objekten wird verdeutlicht, indem die Kommunikation zwischen zwei ausgewählten Objekten grafisch dargestellt wird. In diesem Zusammenhang wird das Nachrichtenkonzept der objektorientierten Programmierung wiederholt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels</p> <p>(a) Analyse der Problemstellung</p> <p>(b) Analyse der Modellierung (Implementationsdiagramm)</p> <p>(c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse)</p> <p>(d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung)</p> <p>(e) Dokumentation von Klassen</p> <p>(f) Implementierung der Anwendung oder von Teilen der Anwendung</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (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), • 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). 	<p><i>Beispiel:</i> Wetthüpfen</p> <p>Für ein Wetthüpfen zwischen einem Hasen, einem Hund und einem Vogel werden die Tiere gezeichnet. Alle Tiere springen wiederholt nach links. Die Höhe und Weite jedes Hüpfers ist zufällig. Evtl. marschieren sie anschließend hintereinander her.</p> <p><i>Materialien:</i></p> <p>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.1-Wiederholung</p>

Unterrichtsvorhaben Q1-II

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, linearen Datenstrukturen

Leitfragen: *Wie können beliebig viele linear angeordnete Daten im Anwendungskontext verwaltet werden?*

Vorhabenbezogene Konkretisierung:

Nach Analyse einer Problemstellung in einem geeigneten Anwendungskontext, in dem Daten nach dem First-In-First-Out-Prinzip verwaltet werden, werden der Aufbau von Schlangen am Beispiel dargestellt und die Operationen der Klasse `Queue` erläutert. Anschließend werden für die Anwendung notwendige Klassen modelliert und implementiert. Eine Klasse für eine den Anforderungen der Anwendung entsprechende Oberfläche sowie die Klasse `Queue` wird dabei von der Lehrkraft vorgegeben. Anschließend wird die Anwendung modifiziert, um den Umgang mit der Datenstruktur zu üben. Anhand einer Anwendung, in der Daten nach dem Last-In-First-Out-Prinzip verwaltet werden, werden Unterschiede zwischen den Datenstrukturen Schlange und Stapel erarbeitet. Um einfacher an Objekte zu gelangen, die zwischen anderen gespeichert sind, wird die Klasse `List` eingeführt und in einem Anwendungskontext verwendet. In mindestens einem weiteren Anwendungskontext wird die Verwaltung von Daten in Schlangen, Stapeln oder Listen vertieft. Modellierungen werden dabei in Entwurfs- und Implementationsdiagrammen dargestellt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Schlange im Anwendungskontext unter Nutzung der Klasse <code>Queue</code></p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse <code>Queue</code></p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse <code>Queue</code> Erstellen und Verwalten größerer Mengen einfacher Objekte</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none">• stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),• erläutern Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (A),• ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),	<p><i>Beispiel:</i> Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger)</p> <p>Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird.</p>

	<ul style="list-style-type: none"> • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • 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). 	<p>Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse Queue.</p> <p>Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen. Objekte werden von zwei Schlangen verwaltet.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange</p>
<p>2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Erarbeitung der Funktionalität der Klasse Stack</p> <p>(c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack</p>		<p><i>Beispiel:</i> Kisten stapeln In einem Stapel nummerierter Kisten soll eine bestimmte Kiste gefunden und an einen Kunden geliefert werden. Dazu müssen Kisten auf verschiedene Stapel gestapelt und wieder zurückgestellt werden.</p>
<p>3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List</p>		<p><i>Beispiel:</i> Abfahrtslauf Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach</p>

<p>(a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen</p> <p>(b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.</p>		<p>ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen</p>
<p>4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext</p>		<p><i>Beispiel:</i> Skispringen Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen</p>

Unterrichtsvorhaben Q1-III

Thema: Suchen und Sortieren auf linearen Datenstrukturen

Leitfragen: *Wie kann man gespeicherte Informationen günstig (wieder-)finden?*

Vorhabenbezogene Konkretisierung:

In einem Anwendungskontext werden zunächst Informationen in einer linearen Liste bzw. einem Feld gesucht. Hierzu werden Verfahren entwickelt und implementiert bzw. analysiert und erläutert, wobei neben einem iterativen auch ein rekursives Verfahren thematisiert wird und mindestens ein Verfahren selbst entwickelt und implementiert wird. Die verschiedenen Verfahren werden hinsichtlich Speicherbedarf und Zahl der Vergleichsoperationen miteinander verglichen.

Anschließend werden Sortierverfahren entwickelt und implementiert (ebenfalls für lineare Listen und Felder). Hierbei soll auch ein rekursives Sortierverfahren entwickelt werden. Die Implementierungen von Quicksort sowie dem Sortieren durch Einfügen werden analysiert und erläutert. Falls diese Verfahren vorher schon entdeckt wurden, sollen sie hier wiedererkannt werden. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

Abschließend werden verschiedene Sortierverfahren hinsichtlich der Anzahl der benötigten Vergleichsoperationen und des Speicherbedarfs beurteilt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Suchen von Daten in Listen und Arrays</p> <ul style="list-style-type: none"> (a) Lineare Suche in Listen und in Arrays (b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen (c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), 	<p><i>Beispiel:</i> Karteiverwaltung Für ein Adressverwaltungsprogramm soll eine Methode zum Suchen einer Adresse geschrieben werden.</p> <p>oder</p> <p><i>Beispiel:</i> Bundesjugendspiele Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken. Im Wettkampfbüro wird das Ergebnis</p>
<p>2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren</p> <ul style="list-style-type: none"> (a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste (b) Implementierung eines einfachen Sortierverfahrens für ein Feld (c) Entwicklung eines rekursiven Sortierverfahrens 		

<p>rens für ein Feld (z.B. Sortieren durch Mischen)</p>	<ul style="list-style-type: none"> • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) , • wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I). 	<p>erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin herausuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können.</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren</p>
<p>3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen</p> <p>(a) <i>Grafische Veranschaulichung der Sortierverfahren</i></p> <p>(b) <i>Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarfs bei beiden Sortierverfahren</i></p> <p>(c) <i>Beurteilung der Effizienz der beiden Sortierverfahren</i></p>		

Unterrichtsvorhaben Q1-IV

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Leitfragen: *Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?*

Vorhabenbezogene Konkretisierung:

Ausgehend von einer vorhandenen Datenbank entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

In anderen Anwendungskontexten müssen Datenbanken erst noch entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in Datenbankschemata überführt. Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden.

Ein Entity-Relationship-Diagramm kann auch verwendet werden, um die Entitäten inklusive ihrer Attribute und Relationen in einem vorgegebenen Datenbankschema darzustellen.

An einem Beispiel wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>(a) Aufbau von Datenbanken und Grundbegriffe</p> <ul style="list-style-type: none"> • Entwicklung von Fragestellungen zur vorhandenen Datenbank • Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema <p>(b) SQL-Abfragen</p> <ul style="list-style-type: none"> • Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle • Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) <p>(c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), • analysieren und erläutern eine Datenbankmodellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • bestimmen Primär- und Sekundärschlüssel (M), • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überführen Datenbankschemata in die 1. bis 3. Normalform (M), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), • nutzen das verfügbare Informatiksystem zur 	<p><i>Beispiel: VideoCenter</i> VideoCenter ist die Simulation einer Online-Videothek für den Informatik-Unterricht mit Webfrontends zur Verwaltung der Kunden, der Videos und der Ausleihe. Außerdem ist es möglich direkt SQL-Abfragen einzugeben. Es ist auch möglich, die Datenbank herunter zu laden und lokal zu installieren. Unter http://dokumentation.videocenter.schule.de/old/video/index.html (abgerufen: 30. 03. 2014) findet man den Link zu dem VideoCenter-System sowie nähere Informationen.</p> <p><i>Beispiel: Schulbuchausleihe</i> Unter http://www.brd.nrw.de/lerntreffs/informatik/structure/material/sek2/datenbanken.php (abgerufen: 30. 03. 2014) wird eine Datenbank zur Verfügung gestellt, die Daten einer Schulbuch-Ausleihe enthält (über 1000 Entleiher, 200 Bücher mit mehreren tausend Exemplaren und viele Ausleihvorgänge). Die Datenbank kann in OpenOffice eingebunden werden.</p>

	<p>strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K).</p>	
<p>2. Modellierung von relationalen Datenbanken</p> <p>(a) Entity-Relationship-Diagramm</p> <ul style="list-style-type: none"> • Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms • Erläuterung und Modifizierung einer Datenbankmodellierung <p>(b) Entwicklung einer Datenbank aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> • Modellierung eines relationalen Datenbankschemas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln <p>(c) Redundanz, Konsistenz und Normalformen</p> <ul style="list-style-type: none"> • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation <p>(d) Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</p>		<p><i>Beispiel: Fahrradverleih</i> Der Fahrradverleih <i>BTR (BikesToRent)</i> verleiht unterschiedliche Typen von Fahrrädern diverser Firmen an seine Kunden. Die Kunden sind bei <i>BTR</i> registriert (Name, Adresse, Telefon). <i>BTR</i> kennt von den Fahrradfirmen den Namen und die Telefonnummer. Kunden von <i>BTR</i> können CityBikes, Treckingräder und Mountainbikes ausleihen.</p> <p><i>Beispiel: Reederei</i> Die Datenverwaltung einer Reederei soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird.</p> <p><i>Beispiel: Buchungssystem</i> In dem Online-Buchungssystem einer Schule können die Lehrer Medienräume, Beamer, Laptops, Kameras, usw. für einen bestimmten Zeitpunkt buchen, der durch Datum und die Schulstunde festgelegt ist.</p>

		<p>Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden.</p> <p>Unter http://mrbs.sourceforge.net (abgerufen: 30.03. 2014) findet man ein freies Online-Buchungssystem inklusive Demo, an Hand derer man erläutern kann, worum es in dem Projekt geht.</p> <p><i>Beispiel: Schulverwaltung</i></p> <p>In einer Software werden die Schulhalbjahre, Jahrgangsstufen, Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, dass z.B. Schüler X von Lehrer Y im 2. Halbjahr des Schuljahrs 2011/2012 in der Jahrgangsstufe 9 im Differenzierungsbereich im Fach Informatik die Note „sehr gut“ erhalten hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden.</p>
--	--	---

Unterrichtsvorhaben Q1-V

Thema: Sicherheit und Datenschutz in Netzstrukturen

Leitfragen: *Wie werden Daten in Netzwerken übermittelt? Was sollte man in Bezug auf die Sicherheit beachten?*

Vorhabenbezogene Konkretisierung:

Anschließend an das vorhergehende Unterrichtsvorhaben zum Thema Datenbanken werden der Datenbankzugriff aus dem Netz, Topologien von Netzwerken, eine Client-Server-Struktur, das TCP/IP-Schichtenmodell sowie Sicherheitsaspekte beim Zugriff auf Datenbanken und verschiedene symmetrische und asymmetrische kryptografische Verfahren analysiert und erläutert. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht runden das Unterrichtsvorhaben ab.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken</p> <p>(a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs</p> <p>(b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz</p> <p>(c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • 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 Interessenla- 	<p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken</p>
		<p><i>Materialien:</i></p>

2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	gen (A), <ul style="list-style-type: none">• nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	<i>Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz</i>
---	--	---

<p><u>Unterrichtsvorhaben Q2-I</u> <i>Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Implementieren • Darstellen und Interpretieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache <p>Zeitbedarf: 24 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u> <i>Endliche Automaten und formale Sprachen</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Endliche Automaten und formale Sprachen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: 20 Stunden</p>
<p><u>Unterrichtsvorhaben Q2-III</u> <i>Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit</i></p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> • Argumentieren • Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> • Informatiksysteme • Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> • Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung <p>Zeitbedarf: 12 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-IV</u> <i>Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase</i></p>

Unterrichtsvorhaben Q2-I

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen

Leitfragen: *Wie können Daten im Anwendungskontext mit Hilfe binärer Baumstrukturen verwaltet werden? Wie kann dabei der rekursive Aufbau der Baumstruktur genutzt werden? Welche Vor- und Nachteile haben Suchbäume für die geordnete Verwaltung von Daten?*

Vorhabenbezogene Konkretisierung:

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt.

Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse `BinaryTree` (der Materialien für das Zentralabitur in NRW) der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum → Inorder-Ausgabe an einem Beispiel verdeutlicht wird.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse `BinarySearchTree` (der Materialien für das Zentralabitur in NRW) weitere Klassen oder Methoden in diesem Anwendungskontext modelliert und implementiert. Auch in diesem Kontext werden grafische Darstellungen der Bäume verwendet.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderen Kontexten weiter geübt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <p>(a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit)</p> <p>(b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), 	<p><i>Beispiel:</i> Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baumstrukturen verdeutlicht.</p> <p>oder</p> <p><i>Beispiel:</i> Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat.</p> <p><i>Weitere Beispiele für Anwendungskontexte für binäre Bäume:</i></p> <p><i>Beispiel:</i> Suchbäume (zur sortierten Speicherung von Daten) Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>oder</p> <p><i>Beispiel:</i> Entscheidungsbäume Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die</p>

	<ul style="list-style-type: none"> • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D). 	<p>Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teilbaum, die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum.</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <p>(a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms</p> <p>(c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen</p> <p>(d) Implementierung der Anwendung oder von Teilen der Anwendung</p> <p>(e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf</p>		<p><i>Beispiel: Informatikerbaum als binärer Baum</i> In einem <i>binären Baum</i> werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum (Material Q2.1.2 abgerufen 21.07.2015)</p>
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <p>(a) Analyse der Problemstellung, Ermittlung</p>		<p><i>Beispiel: Informatikerbaum als Suchbaum</i> In einem binären <i>Suchbaum</i> werden die Namen und die Geburtsdaten von Informatikern</p>

<p>von Objekten, ihren Eigenschaften und Operationen</p> <p>(b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften</p> <p>(c) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation</p> <p>Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums</p>		<p>kern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.)</p> <p>Folgende Funktionalitäten werden benötigt:</p> <ul style="list-style-type: none"> • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum (Material Q2.1.3 abgerufen 21.07.2015)</p>
<p>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>		<p><i>Beispiel:</i> Codierungsbäume (s.o.) oder Huffman-Codierung</p>

Unterrichtsvorhaben Q2-II

Thema: Endliche Automaten und formale Sprachen

Leitfragen: *Wie kann man (endliche) Automaten genau beschreiben? Wie können endliche Automaten (in alltäglichen Kontexten oder zu informatischen Problemstellungen) modelliert werden? Wie können Sprachen durch Grammatiken beschrieben werden? Welche Zusammenhänge gibt es zwischen formalen Sprachen, endlichen Automaten und regulären Grammatiken?*

Vorhabenbezogene Konkretisierung:

Anhand kontextbezogener Beispiele werden endliche Automaten entwickelt, untersucht und modifiziert. Dabei werden verschiedene Darstellungsformen für endliche Automaten ineinander überführt und die akzeptierten Sprachen endlicher Automaten ermittelt. An einem Beispiel wird ein nichtdeterministischer Akzeptor eingeführt als Alternative gegenüber einem entsprechenden deterministischen Akzeptor.

Anhand kontextbezogener Beispiele werden Grammatiken regulärer Sprachen entwickelt, untersucht und modifiziert. Der Zusammenhang zwischen regulären Grammatiken und endlichen Automaten wird verdeutlicht durch die Entwicklung von allgemeinen Verfahren zur Erstellung einer regulären Grammatik für die Sprache eines gegebenen endlichen Automaten bzw. zur Entwicklung eines endlichen Automaten, der genau die Sprache einer gegebenen regulären Grammatik akzeptiert.

Auch andere Grammatiken werden untersucht, entwickelt oder modifiziert. An einem Beispiel werden die Grenzen endlicher Automaten ausgelotet.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Endliche Automaten</p> <p>(a) Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten</p> <p>(b) Untersuchung, Darstellung und Entwicklung endlicher Automaten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), • 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 endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A). 	<p><i>Beispiel:</i> Entwicklung einer Anleitung zur Bedienung von Handys, dabei Einführung der Begriffe Zustand, Eingabe, Übergang.</p>
<p>2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen</p> <p>(a) Erarbeitung der formalen Darstellung regulärer Grammatiken</p> <p>(b) Untersuchung, Modifikation und Entwicklung von Grammatiken</p> <p>(c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden</p> <p>(d) Entwicklung regulärer Grammatiken zu endlichen Automaten</p>		<p><i>Beispiel:</i> Grammatiken zur Erzeugung verschiedener formaler Sprachen, z.B. angelehnt an die deutsche Sprache oder eine Programmiersprache.</p>
<p>3. Grenzen endlicher Automaten</p>		<p><i>Beispiel:</i> Entwicklung einer Grammatik zur Erzeugung von Wörtern der Sprache a^*b^*, wobei jeweils gleich viele a's und b' vorkommen müssen. Begründung, dass es keinen endlichen Automaten zu dieser Sprache geben kann.</p>

Unterrichtsvorhaben Q2-III

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Leitfragen: *Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinenahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen?*

Vorhabenbezogene Konkretisierung:

Anhand einer von-Neumann-Architektur und einem maschinennahen Programm wird die prinzipielle Arbeitsweise von Computern verdeutlicht.

Ausgehend von den prinzipiellen Grenzen endlicher Automaten liegt die Frage nach den Grenzen von Computern bzw. nach Grenzen der Automatisierbarkeit nahe. Mit Hilfe einer entsprechenden Java-Methode wird plausibel, dass es unmöglich ist, ein Informatiksystem zu entwickeln, das für jedes beliebige Computerprogramm und jede beliebige Eingabe entscheidet ob das Programm mit der Eingabe terminiert oder nicht (Halteproblem). Anschließend werden Vor- und Nachteile der Grenzen der Automatisierbarkeit angesprochen und der Einsatz von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen beurteilt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme</p> <p>a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher</p> <p>b) einige maschinennahe Befehle und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann</p> <p>c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), • untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A). 	<p><i>Beispiel:</i> Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Programmierung</p>
<p>2. Grenzen der Automatisierbarkeit</p> <p>a) Vorstellung des Halteproblems</p> <p>b) Unlösbarkeit des Halteproblems</p> <p>c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen</p>		<p><i>Beispiel:</i> Halteproblem</p> <p><i>Materialien:</i> Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem</p>

Unterrichtsvorhaben Q2-IV

Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahrs der Qualifikationsphase

c. Die Qualifikationsphase (Q1 und Q2) - Leistungskurs

Bei einer ausreichender Anzahl an Interessenten ist in seltenen Fällen auch ein Leistungskurs im Fach Informatik möglich. Die Inhalte orientieren sich dabei im Wesentlichen an den Unterrichtssequenzen des Grundkurses, werden aber um folgende Kompetenzen erweitert¹:

- Daten und ihre Strukturierung
 - Objekte und Klassen
 - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M).
 - Datenbanken
 - implementieren ein relationales Datenbankschema als Datenbank (I).
- Algorithmen
 - Analyse, Entwurf und Implementierung von Algorithmen
 - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ und „Backtracking“ (M),
 - testen Programme systematisch anhand von Beispielen und mit Hilfe von Testanwendungen (I).
 - Algorithmen in ausgewählten informatischen Kontexten
 - erläutern Operationen dynamischer (linearer und nicht-linearer) Datenstrukturen (A),
 - implementieren Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (I),
 - implementieren und erläutern iterative und rekursive Such- und Sortierverfahren unterschiedlicher Komplexitätsklassen (Speicherbedarf und Laufzeitverhalten) (I),
 - erläutern das Prinzip der Nebenläufigkeit (A),
 - analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A),
 - entwickeln und implementieren Algorithmen und Methoden zur Client-Server-Kommunikation (I).
- Formale Sprachen und Automaten
 - Endliche Automaten
 - analysieren und erläutern die Eigenschaften endlicher Automaten und Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A),
 - ermitteln die Sprache, die ein endlicher Automat oder ein Kellerautomat akzeptiert (D),
 - entwickeln und modifizieren zu einer Problemstellung endliche Automaten oder Kellerautomaten (M),
 - entwickeln zur Grammatik einer regulären oder kontextfreien Sprache einen zugehörigen endlichen Automaten oder einen Kellerautomaten (M).
 - Grammatiken regulärer Sprachen
 - analysieren und erläutern Grammatiken regulärer und kontextfreier Sprachen (A),
 - modifizieren Grammatiken regulärer und kontextfreier Sprachen (M),

¹ Veränderungen gegenüber dem Grundkurs sind rot markiert.

- entwickeln zu einer regulären oder kontextfreien Sprache eine Grammatik, die die Sprache erzeugt (M).
- Scanner, Parser und Interpreter für eine reguläre Sprache
 - modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I).
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen
 - erläutern die Grenzen endlicher Automaten und regulärer Sprachen im Anwendungszusammenhang auf (A).
- Informatiksysteme
 - Einzelrechner und Rechnernetzwerke
 - analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A),
 - entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M).
 - Nutzung von Informatiksystemen
 - nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Daten, zur Organisation von Arbeitsabläufen sowie zur Verteilung und Zusammenführung von Arbeitsanteilen (K),
 - Wenden didaktisch orientierte Entwicklungsumgebungen zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),
 - entwickeln mit didaktisch orientierten Entwicklungsumgebungen einfache Benutzungsoberflächen zur Kommunikation mit einem Informatiksystem (M).
 - Sicherheit
 - analysieren und erläutern Eigenschaften, Funktionsweisen und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).

III. Grundsätze der Leistungsbewertung im Fach Informatik

Die Leistungsbewertung beruht auf den im Schulgesetz NRW §48 festgelegten Grundsätzen, den in den Vorgaben der für das Zentralabitur vorliegenden Bewertungsregeln und den entsprechenden Beschlüssen der Fachkonferenz.

Die Leistungsbewertung stellt in pädagogischer und gesellschaftlicher Hinsicht für alle am Lernprozess Beteiligten eine wichtige Rückmeldung dar. Insbesondere hilft sie den Schülerinnen und Schülern bei der Wahrnehmung der eigenen Position und damit der Schaffung eines realistischen Selbstbildes. Für die Erziehungsberechtigten hat die Leistungsbewertung eine besondere Rückmeldefunktion. Die Lehrerinnen und Lehrer können aus dem Leistungsbild für sich wichtige Erkenntnisse zum Unterrichtsfolg und zur Unterrichtsgestaltung gewinnen.

Schülerinnen und Schülern, die erkennbar die den geforderten Leistungsstandard nicht erbringen, werden aufgrund einer Diagnose der individuellen Schwächen Förderhinweise gegeben bzw. es werden mit ihnen und ihren Eltern Förderungsmaßnahmen vereinbart. Der momentane Leistungsstand wird auf Wunsch zeitnah, auf jeden Fall aber gegen Quartalsende mitgeteilt. Beratungsgespräche mit dem Ziel der individuellen Förderung sind stets in den Sprechstunden der Lehrkräfte und auf dem Elternsprechtag möglich.

Zur Leistungsbewertung werden die

- schriftlichen Arbeiten
- mündlichen Beiträge
- praktischen Leistungen

herangezogen. Bewertungsgrundlagen sind zum einen das im Unterricht erarbeitete und vermittelte Wissen und zum anderen die erlernten Fertigkeiten und Kompetenzen.

Bewertet werden dabei

- der Umfang und die Vollständigkeit
- der Grad der Korrektheit und der Sicherheit
- das Maß der Selbstständigkeit und Kreativität
- der Gebrauch der Fachsprache und die Verwendung fachspezifischer Darstellungsformen.

Zur Sicherung eines einheitlichen Standards bei Inhalten und den Anforderungsprofilen sprechen sich die Fachkolleginnen und –kollegen bei Parallelunterricht ständig ab und vereinbaren einen regelmäßigen Austausch von Arbeitsmaterialien, Arbeitsergebnissen und Klausurtexten. Dabei werden insbesondere inhaltliche und methodische Schwerpunkte festgelegt und die zugehörigen Bewertungskriterien vereinbart.

Im Interesse einer größtmöglichen Transparenz werden das jeweilig gültige Fachcurriculum und die aktuellen Festlegungen zur Leistungsbewertung auf der Homepage der Schule veröffentlicht.

IV. Formen der Leistungsbewertung

Das Fach Informatik eignet sich in besonderem Maße für eine projekt- und produktorientierte Unterrichtsgestaltung. Die Bewertung der erbrachten Leistungen lässt sich im Wesentlichen auf die Bereiche schriftliche Leistungsüberprüfungen und sonstige Mitarbeit aufteilen.

1. Beurteilungsbereich schriftliche Lernerfolgsüberprüfungen

a. Klausuren

Anzahl und Dauer der Klausuren

Laut Konferenzbeschluss werden die Klausuren im Fach Informatik wie folgt verteilt:

Jahrgangsstufe	Anzahl	Dauer (in Min)	Hinweis
8	2 je Halbjahr*	45	Alle SuS des Kurses
9	2 je Halbjahr*	45	

* Eine Klausur je **Schuljahr** kann nach Maßgabe der Lehrkraft durch eine Projektarbeit ersetzt werden.

EF	1 (1.Hj) 2 (2.Hj)	90	Nur für SuS, die dieses Fach als schriftliches Fach gewählt haben.
GK Q1	2 je Halbjahr ²	90	
GK Q2	2 (1.Hj) 1 (2.Hj)	135 225	

LK Q1	2 je Halbjahr ³	135	
LK Q2	2 (1.Hj) 1 (2.Hj)	225 270	

Aufgabenstellungen innerhalb von Klausuren

Die im Rahmen von schriftlichen Leistungsüberprüfungen dieser Art gewählten Aufgabenstellungen sollen die Vielfalt der im Unterricht erworbenen inhaltlichen und methodischen Kompetenzen widerspiegeln. Den Vorgaben für das Abitur üblichen Regelungen entsprechend, wird ein Teil der Aufgaben dem Anforderungsbereich I - Reproduktion von Kenntnissen entnommen. Schülerinnen und Schüler sollen allerdings in einem angemessenen Maß auch Aufgaben bearbeiten, bei denen es bspw. um die Analyse von Algorithmen, Herleiten von Zusammenhängen, Vergleichen sowie Modellieren und Implementieren geht (Anforderungsbereich II - Reorganisation und Transfer von Wissen). Darüber hinaus werden Aufgaben gestellt, die eine kritische Reflexion neuer Sachverhalte fordern (Anforderungsbereich III).

² Sollte, die am Städt. Gymnasium Leichlingen obligatorische, Facharbeit in Jgst. Q1 im Fach Informatik angefertigt werden, reduziert sich die Anzahl der Klausuren im zweiten Halbjahr auf eine.

³ Siehe Fußnote 2

Die Einbindung sämtlicher Anforderungsbereiche trägt wesentlich dazu bei, die in den Bildungsstandards und Vorgaben geforderten inhalts- und prozessorientierten Kompetenzen valide zu überprüfen, sowie die Bewertungen der erbrachten Prüfungsleistungen für alle Beteiligten transparent zu machen.

Die Fachschaft Informatik verständigt darauf ebenfalls offene Problemstellungen einzubeziehen, bei denen nicht von vornherein eine eindeutige Lösungsstrategie und –form feststehen, und demzufolge Schülerinnen und Schüler individuelle Gestaltungsideen einbringen können.

Die Grundlage zur Zuordnung von erreichten Punkten in der Leistungsüberprüfung zu Noten bzw. Punkten ist die Verteilung in den Abiturvorgaben. In der Sekundarstufe I kann auf Grund pädagogischer Erwägungen in Einzelfällen von dieser stringenten Verteilung abgewichen werden.

2. Beurteilungsbereich sonstige Mitarbeit

Im Rahmen dieses Beurteilungsbereichs werden die Qualität und Kontinuität der Beiträge, die die Schülerinnen und Schüler im Unterricht einbringen, erfasst. Leistungen im Bereich der sonstigen Mitarbeit können in mündlicher, schriftlicher oder praktischer Form erbracht werden und sollen in Anbindung an die Problemstellungen, die inhaltliche Reichweite ausschöpfen und dem Anspruchsniveau der jeweiligen Unterrichtsreihe entsprechen.

Der Bereich sonstige Mitarbeit umfasst alle Leistungen, die nicht in Form einer Klausur oder im Rahmen eines schriftlichen Kurztests erbracht werden. Beurteilungsrelevant sind dabei u.a. (in alphabetischer Reihenfolge):

- Angekündigte schriftliche Kurztests (Dauer: max. 20 Minuten) über Inhalte des letzten Unterrichtsabschnitts
- Beteiligung an Unterrichtsgesprächen
- Dokumentation der erstellten Programme
- Entwicklung eigener Ideen und Lösungsstrategien und deren Darstellung
- Erläuterung und Demonstration von Lösungen und Lösungsansätzen
- Implementierungen in der jeweiligen Programmiersprache
- Kompetenter Gebrauch der Fachsprache
- Kritikfähigkeit – auch im Umgang mit fremden Ideen
- Präsentation von Arbeitsergebnissen
- Programmiersprachenunabhängige Problemstrukturierung
- schriftliche ad hoc Überprüfung von Hausaufgaben bzw. aktuellen Inhalten
- Umgang mit der im Unterricht eingesetzten Software
- Umgang mit der zur Verfügung gestellten Rechnerkonfiguration
- Verteidigung von Ideen in fachgerechter und angemessener Form
- Wiederholungen vorangegangenen Stoffs

a. Kriterien und Indikatorenkatalog zur Bewertung der sonstigen Mitarbeit

Die einleitend angesprochene Transparenz der Leistungsbewertung gegenüber den Schülerrinnen und Schülern sieht die Fachschaft Informatik als einen Eckpfeiler gerechter Leistungsbeurteilung. Die Maßstäbe und Instrumentarien zur Bewertung zu entwickeln und zu vereinheitlichen ist ein Ziel der Qualitätsentwicklung unserer Schule. Zu diesem Zwecke verständigen sich die Lehrkräfte über den Kriterien- und Indikatorenkatalog im Anhang I, der in dieser Form auch allen Schülerrinnen und Schülern bekannt gemacht und zur Verfügung gestellt wird.

3. Verhältnis schriftliche Leistungsüberprüfungen zu sonstiger Mitarbeit / Gesamtnote

Eine rein rechnerische Notenmittlung zwischen den schriftlichen und sonstigen Leistungen soll nicht erfolgen, pädagogischen Grundsätzen ist der Vorrang vor einer rein arithmetischen Notenberechnung einzuräumen.

Unter Beachtung der Anzahl und Dauer der jeweiligen schriftlichen Leistungen soll in allen Jahrgangsstufen die sonstige Mitarbeit und die schriftlichen Leistungen zu gleichen Anteilen in die Endnote einfließen.

V. Anhang I: Kriterien- und Indikatorenkatalog zur Leistungsbewertung

Leistungsbewertung im Fach Informatik	Häufigkeit der Mitarbeit	Qualität der Mitarbeit	Beherrschung der Fachmethoden und der Fachsprache	Zusammenarbeit im Team	Präsentation von Referaten, Protokollen u. a.	Vor- und Nachbereitung des Unterrichts / Bereitstellung der AM /Heftführung
sehr gut Die Leistung entspricht den Anforderungen in besonderem Maße.	Ich arbeite in jeder Stunde regelmäßig mit.	Ich kann Gelerntes sicher wiedergeben und anwenden. Oft finde ich auch neue Lösungswege und Ideen.	Ich kann die gelernten Methoden sehr sicher anwenden. Die Fachsprache beherrsche ich sehr gut.	Ich höre immer genau zu, gehe sachlich auf andere ein, ergreife bei der Arbeit die Initiative.	Ich bin sehr häufig und freiwillig bereit, Arbeitsergebnisse vorzustellen (, sowie gegebenenfalls Referate in den Unterricht einzubringen)	Ich führe mein Informatikheft kontinuierlich, übersichtlich und sorgfältig. Ich habe immer alle Arbeitsmaterialien mit, mache immer die Hausaufgaben, beginne stets pünktlich mit der Arbeit.
gut Die Leistung entspricht voll den Anforderungen.	Ich arbeite in der Mehrzahl der Stunden regelmäßig mit.	Ich kann Gelerntes sicher wiedergeben und anwenden. Manchmal finde ich auch neue Lösungswege und Ideen.	Ich kann die gelernten Methoden meist sicher anwenden. Die Fachsprache beherrsche ich gut.	Ich höre meistens zu, gehe sachlich auf andere ein, kann mit anderen erfolgreich an einer Sache arbeiten.	Ich bin häufig und freiwillig bereit, Arbeitsergebnisse vorzustellen (, sowie gegebenenfalls Referate in den Unterricht einzubringen).	Ich führe mein Informatikheft in der Regel kontinuierlich, übersichtlich und sorgfältig. Ich habe fast immer alle Arbeitsmaterialien mit, mache fast immer die Hausaufgaben, beginne fast immer pünktlich mit der Arbeit.
befriedigend Die Leistung entspricht im Allgemeinen den Anforderungen.	Ich arbeite häufig mit.	Ich kann Gelerntes wiedergeben und meist auch anwenden. Neue Lösungswege suche ich kaum.	Ich kann die gelernten Methoden vom Prinzip her anwenden. Die Fachsprache beherrsche ich im Wesentlichen.	Ich höre häufig zu, gehe sachlich auf andere ein, kann mit anderen an einer Sache arbeiten.	Ich bin manchmal oder nach Aufforderung bereit, Arbeitsergebnisse vorzustellen (, sowie gegebenenfalls Referate in den Unterricht einzubringen).	Ich führe mein Informatikheft in der Regel übersichtlich und sorgfältig. Ich habe meistens alle Arbeitsmaterialien mit, mache meistens die Hausaufgaben, beginne meist pünktlich mit der Arbeit.
ausreichend Die Leistung zeigt Mängel, entspricht im Ganzen jedoch den Anforderungen.	Ich arbeite nur selten freiwillig mit, ich muss meistens aufgefordert werden.	Ich kann Gelerntes grob wiedergeben, aber nicht immer an anderen Beispielen anwenden.	Ich kann die gelernten Methoden nicht immer anwenden. Die Fachsprache beherrsche ich nur wenig.	Ich höre häufiger nicht zu und gehe nicht immer auf andere ein. Ich arbeite nur wenig erfolgreich mit anderen zusammen.	Ich bin selten bereit selbstständig Arbeitsergebnisse vorzustellen (, sowie gegebenenfalls Referate in den Unterricht einzubringen).	Ich führe mein Informatikheft. Ich habe häufiger alle Arbeitsmaterialien mit, mache meistens die Hausaufgaben, beginne oft pünktlich mit der Arbeit.
mangelhaft Die Leistung entspricht nicht den Anforderungen. Grundkenntnisse sind vorhanden. Mängel können in absehbarer Zeit behoben werden.	Ich arbeite ganz selten freiwillig mit, ich muss fast immer aufgefordert werden.	Ich kann Gelerntes nur mit Lücken oder falsch wiedergeben. Auf andere Beispiele kann ich es fast nie anwenden.	Ich kann die gelernten Methoden kaum anwenden. Die Fachsprache beherrsche ich nicht.	Ich höre kaum zu, gehe nur selten auf andere ein, arbeite sehr ungern mit anderen zusammen.	Ich bringe Referate und Arbeitsergebnisse fast überhaupt nicht in den Unterricht ein	Ich führe mein Informatikheft lückenhaft. Ich habe häufiger (unvollständige) Arbeitsmaterialien mit, mache oft die Hausaufgaben, beginne gewöhnlich erst nach Aufforderung mit der Arbeit.

(Natürlich gibt es im Fach Informatik auch die Note ‚ungenügend‘, wenn die Leistung den Anforderungen nicht entspricht und auch die Grundkenntnisse so lückenhaft sind, dass die Mängel in absehbarer Zeit nicht behoben werden können. Das sollte sich doch wohl vermeiden lassen – oder?!)

verändert nach: <http://www.huma-gym.de>, 02/2013