

# Schulinterner Lehrplan Informatik Sekundarstufe I

*Mit Verweis auf die Kompetenzen des Kernlehrplans*



*Fachschaft Informatik*



# Inhalt

Informatik am B.M.V.-Gymnasium.....	1
Lehr- und Lernmittel .....	1
Lehrmittel.....	1
Medienkonzept .....	1
Entscheidung über die eingesetzten Programmiersprachen und Softwareprodukte.....	1
Unterrichtsvorhaben.....	2
Klasse 6.....	3
Differenzierungskurs Klasse 9 und 10 .....	7
Grundsätze der Leistungsbewertung .....	11
Klasse 6.....	11
Wahlpflichtbereich Klasse 9 und 19 .....	11
Schriftliche Arbeiten.....	11
Sonstige Mitarbeit .....	12
Individuelle Förderung und Differenzierungsmaßnahmen .....	12
Differenzierung nach Lerntempo .....	13
Differenzierung durch offene Aufgabenstellungen.....	13
Förderung außerhalb des Unterrichts.....	13
Förderung von Mädchen in der Informatik.....	13
Literatur.....	14



## Informatik am B.M.V.-Gymnasium

In einer Welt, die immer stärker von Digitalisierung geprägt wird, haben informatische Phänomene Einzug in die Lebens- und Arbeitswelt der modernen Gesellschaft gehalten. Aufgabe des Informatikunterrichts ist es, Kinder und Jugendliche bestmöglich auf ein Leben in dieser Welt vorzubereiten, ihnen also eine **informatische Grundbildung** zu vermitteln. Dazu wird Informatikunterricht in allen Phasen der schulischen Laufbahn angeboten:

- **Klasse 6**                    Verpflichtend für alle Schülerinnen und Schüler (Zwei Wochenstunden à 67,5 Minuten)
- **Klasse 9 & 10**            Als zweijähriges Wahlfach im Differenzierungsbereich in Klasse 9 und 10
- **Sekundarstufe II**        Wählbar sowohl als Grund- als auch als Leistungskurs

In der Erprobungsstufe (Klasse 6) lernen die Schülerinnen und Schüler Grundprinzipien aus verschiedenen Teilbereichen der Informatik kennen. Dabei liegt das Augenmerk verstärkt auf dem sinnvollen, produktiven und selbstständigen Einsatz von Informatiksystemen auch unter Berücksichtigung von Sicherheits- und Datenschutzaspekten. Im Differenzierungsbereich in Klasse 9 und 10 werden diese Fähigkeiten vertieft. Lebensweltliche und alltagsrelevante Aspekte stehen nach wie vor im Fokus, diese werden aber zunehmend mit fachlichen Inhalten verwoben. So steht etwa der Umgang mit ersten, formalen Programmiersprachen und Grundlagen der theoretischen Informatik im Fokus. Der Unterricht in der Sekundarstufe II befähigt schließlich, informatische Methoden und Werkzeuge (Datenbanken, Algorithmen, etc.) zum Lösen komplexer Problemstellungen zu nutzen. Insbesondere der Leistungskurs bietet damit eine intensive Vorbereitung auf eine weitere akademische oder berufliche Laufbahn im Bereich der Informatik.

### Lehr- und Lernmittel

#### *Lehrmittel*

Als Lehrwerk steht in der Jahrgangsstufe 6 *Praxis Informatik 5/6* des Westermann Verlags zur Verfügung. Im Differenzierungsbereich kann zur Einführung in die Programmiersprache Python in Klasse 10 *Einfach Informatik* aus dem Klett Verlag genutzt werden.

In der Jahrgangsstufe 9 steht kein Lehrwerk zur Verfügung. Hier wird das benötigte Material in digitaler oder in Papierform zur Verfügung gestellt. In der Sekundarstufe II wird ebenfalls auf die Nutzung eines zentralen Schulbuchs verzichtet. Themenabhängig kann die Lehrperson so auf die Ressourcen verschiedener Verlage sowie Online-Angebote zurückgreifen.

#### *Medienkonzept*

Allen Schülerinnen und Schülern wird ein digitales Endgerät zur Verfügung gestellt. Dabei handelt es sich um ein zentral verwaltetes Windowsgerät, auf dem alle für den Informatikunterricht relevanten Softwareprodukte vorinstalliert sind.

Dies bietet optimale Voraussetzungen für den Einsatz des Geräts im Informatikunterricht. In der Erprobungsstufe kann der Umgang mit diesem auch im Klassenverband eingeübt werden, ohne dass durch verschiedene Betriebssysteme oder Plattformen Unklarheiten entstehen. Auch im weiteren Verlauf der Schullaufbahn wird digitales Arbeiten effizient gestaltet, da Software bereits vorinstalliert werden kann und keine Kompatibilitätsprobleme oder ähnliches auftreten.

#### *Entscheidung über die eingesetzten Programmiersprachen und Softwareprodukte*

Um die Inhalte des Informatikunterrichts bearbeiten zu können ist immer wieder der Einsatz digitaler Ressourcen notwendig. Dabei ist zunächst die Entscheidung über die eingesetzten Programmiersprachen relevant:

In Klasse 6 wird als visuelle, blockbasierte Programmiersprache *Scratch* verwendet. Diese kann direkt im Webbrowser aufgerufen werden und bietet eine deutschsprachige Umgebung zum Einstieg in die Programmierung. Als formale Programmiersprache wird in Klasse 10 das in der Industrie immer häufiger genutzte *Python* eingeführt. Den Vorgaben des Zentralabiturs entsprechend liegt der Fokus in der Sekundarstufe II schließlich auf *Java*.



Nicht nur der Einsatz von Programmiersprachen, sondern auch andere Inhalte setzen den Einsatz von Software voraus. Im Informatikunterricht können dazu die folgenden Produkte genutzt werden:

Jahrgangsstufe	Inhalt	Vorschlag
Klasse 6	Nutzen von Textverarbeitung und Webbrowser	Microsoft Word Microsoft Edge
	Grafische Programmierung mit einer blockbasierten Programmiersprache	Scratch
Differenzierungsbereich	Tabellenkalkulationen	Microsoft Excel
	Logische Schaltungen	LogiSim <a href="#">CircuitVerse</a>
	HTML und CSS	Notepad++
	Programmieren mit Python	<a href="#">TigerJython</a>
Sekundarstufe II	Programmieren mit Java	<a href="#">JavaEditor</a>
	Datenbanken und SQL	OpenOffice BASE <a href="#">XAMPP</a>

Alle eingesetzte Software ist frei erhältlich oder werden seitens der Schule zur Verfügung gestellt. Die hier angeführten Produkte sind nicht als verpflichtende Vorgabe, sondern als Orientierung zu verstehen.

## Unterrichtsvorhaben

Die Übersicht dient dazu, einen schnellen Überblick über Themen bzw. Fragestellungen des Informatikunterrichtes in den einzelnen Jahrgangsstufen zu gewinnen. Die Angabe der einzelnen Unterrichtsvorhaben, der darin thematisierten Inhalte und angestrebten Kompetenzerweiterungen verdeutlicht, welches Wissen und welche Fähigkeiten im Rahmen des Unterrichts zu erlernen sind. Die Spalte *Unterrichtsvorhaben und konkrete Umsetzung* gibt einen Einblick in die Realisierung der inhaltlichen Aspekte im Unterricht unter Angabe konkreter Fragestellungen und Inhalte. Die beiden Spalten *inhaltliche Schwerpunkte* und *konkretisierte Kompetenzerwartungen* orientieren sich an den Kernlehrplan des Landes Nordrhein-Westfalen vorgegebenen Rahmenbedingungen. Jede konkrete Kompetenzerwartung ist einem von vier Kompetenzbereiche zugeordnet, die der Kernlehrplan wie folgt vorgibt:

- Argumentieren (A)
- Modellieren und Implementieren (MI)
- Darstellen und Interpretieren (DI)
- Kommunizieren und Kooperieren

In der rechten Spalte *Hinweise zur Umsetzung* finden sich beispielsweise Vorschläge für einsetzbare Softwareprodukte oder inhaltliche Alternativen. Ein grober zeitlicher Rahmen für die einzelnen Unterrichtsvorhaben findet sich jeweils hinter deren Titel in der Spalte *Unterrichtsvorhaben und konkrete Umsetzung* (jeweils angegeben in Unterrichtseinheiten [UE] à 67,5 Minuten). Auch dieser Zeitrahmen ist lediglich als Orientierung zu verstehen.



## Klasse 6

Da Informatik in der Erprobungsstufe (Klasse 5 und 6) erst seit kurzer Zeit Teil der verpflichtenden Schullaufbahn an Gymnasien ist, unterliegt der Unterricht in diesen Jahrgangsstufen einem steten Wandel. Die aufgeführten Unterrichtsvorhaben und deren Reihenfolge ist deshalb zwar als Orientierung zu verstehen, jedoch unterliegt die konkrete Umsetzung stets dem Ermessen der Lehrperson. Diese kann auf die individuelle Lerngruppe und kurzfristige Veränderungen (etwa durch Einsatz einer neuen Blockbasierten Programmiersprache) eingehen und gegebenenfalls deutlich von dem hier angeführten Ablauf abweichen.

Unterrichtsvorhaben und konkrete Umsetzung	Inhaltliche Schwerpunkte	Konkretisierte Kompetenzerwartungen	Hinweise zur Umsetzung
<p><b>I. Begegnung mit der digitalen Welt (ca. 5 UE)</b></p> <ul style="list-style-type: none"> <li>▪ Was ist Informatik</li> <li>▪ Informatische Phänomene im Alltag erkunden</li> <li>▪ Digitale Medien als Werkzeug zur Datenverarbeitung</li> <li>▪ Begriffsklärungen: Informationen und Daten</li> <li>▪ EVA - Prinzip</li> </ul>	<p><b>Information und Daten</b></p> <ul style="list-style-type: none"> <li>▪ Daten und ihre Codierung</li> <li>▪ Informationsgehalt von Daten</li> </ul> <p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>▪ Aufbau und Funktionsweise von Informatiksystemen</li> </ul> <p><b>Informatik, Mensch und Gesellschaft</b></p> <ul style="list-style-type: none"> <li>▪ Informatiksysteme in der Lebens- und Arbeitswelt</li> </ul>	<p><i>Die Schülerinnen und Schüler...</i></p> <p><b>Information und Daten</b></p> <ul style="list-style-type: none"> <li>▪ erläutern den Datenbegriff anhand von Beispielen aus ihrer Erfahrungswelt (A),</li> <li>▪ erläutern den Zusammenhang und die Bedeutung von Information und Daten (A),</li> <li>▪ interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI),</li> <li>▪ stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI).</li> </ul> <p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>▪ benennen Beispiele für (vernetzte) Informatiksysteme aus ihrer Erfahrungswelt (DI),</li> <li>▪ benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI),</li> <li>▪ beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung (DI).</li> </ul>	



		<p><b>Informatik, Mensch und Gesellschaft</b></p> <ul style="list-style-type: none"> <li>beschreiben an Beispielen die Bedeutung von Informatiksystemen in der Lebens- und Arbeitswelt (KK).</li> </ul>	
<p><b>II. Umgang mit Informatiksystemen am Beispiel der Nutzung der Schulrechner (ca. 6 UE)</b></p> <ul style="list-style-type: none"> <li>Ein- und Ausschalten der Rechner</li> <li>Nutzen der Grundfunktionen (W - Lan, Internetbrowser etc.)</li> <li>Einrichten der Moodle – App</li> <li>Herunterladen von Dateien</li> <li>Anlegen und Umgang mit Dateien und Ordern             <ul style="list-style-type: none"> <li>Aufbau eines Ordnersystems</li> <li>Verschieben, Löschen und Kopieren</li> <li>Dateien anlegen / Dateitypen</li> </ul> </li> </ul>	<p><b>Information und Daten</b></p> <ul style="list-style-type: none"> <li>Daten und ihre Codierung</li> </ul> <p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>Anwendung von Informatiksystemen</li> </ul>	<p><b>Information und Daten</b></p> <ul style="list-style-type: none"> <li>erläutern Einheiten von Datenmengen (A / KK),</li> <li>vergleichen Datenmengen hinsichtlich ihrer Größe mithilfe anschaulicher Beispiele aus ihrer Lebenswelt (DI).</li> </ul> <p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>vergleichen Möglichkeiten der Datenverwaltung hinsichtlich ihrer spezifischen Charakteristika (u. a. Speicherort, Kapazität, Aspekte der Datensicherheit) (A),</li> <li>setzen zielgerichtet Informatiksysteme zur Verarbeitung von Daten ein (MI),</li> <li>erläutern Prinzipien der strukturierten Dateiverwaltung (A),</li> <li>setzen Informatiksysteme zur Kommunikation und Kooperation ein (KK).</li> </ul>	<p>Die Unterrichtsvorhaben II und III können mit dem Ziel einen ‚PC – Führerschein‘ zu erwerben, behandelt werden. Hier bietet sich nach Abschluss des Unterrichtsvorhabens III ein Test an, nach dessen Bestehen den Schülerinnen und Schülern eine entsprechende Urkunde ausgehändigt wird.</p> <p>Als Textverarbeitungsprogramm wird Microsoft-Word genutzt.</p>
<p><b>III. Nutzen von Anwendersoftware I – Textverarbeitung und Webbrowser (ca. 4 UE)</b></p> <p><i>Umgang mit Textverarbeitung</i></p> <ul style="list-style-type: none"> <li>Textdokumente erstellen und abspeichern</li> <li>Markieren und Formatieren</li> <li>Ausrichten von Absätzen</li> <li>Einfügen von Bildern</li> </ul> <p><i>Recherchieren im Internet</i></p> <ul style="list-style-type: none"> <li>Vergleich von Suchmaschinen</li> <li>Beurteilung von Quellen</li> <li>Lizenzrecht bei der Bildrecherche</li> </ul>			
<p><b>IV. Codieren und Chiffren (ca. 12 UE)</b></p>	<p><b>Information und Daten</b></p> <ul style="list-style-type: none"> <li>Daten und ihre Codierung</li> <li>Verschlüsselungsverfahren</li> </ul>	<p><b>Information und Daten</b></p>	<p>Es bieten sich verschiedene, browserbasierte Werkzeuge an, die den Aufwand bei Codierungen und Chiffrierung reduzieren:</p>



<p>Begriffliche Abgrenzung: Codierung und Chiffrierung, Kommunikation</p> <p><i>Codierung</i></p> <ul style="list-style-type: none"> <li>▪ Winkeralphabet</li> <li>▪ Morsealphabet</li> <li>▪ Dualsystem</li> <li>▪ ASCII</li> <li>▪ <i>Optionale: Bildcodierung</i></li> </ul> <p><i>Chiffrierung</i></p> <ul style="list-style-type: none"> <li>▪ Transpositionschiffre (Skytale)</li> <li>▪ Monoalphabetische Chiffren (Cäsar-Chiffre, Monochiffre)</li> <li>▪ Häufigkeitsanalyse</li> <li>▪ <i>Optional: Vigenère-Chiffre</i></li> <li>▪ <i>Optional: Enigma</i></li> </ul>		<ul style="list-style-type: none"> <li>▪ nennen Beispiele für die Codierung von Daten aus ihrer Erfahrungswelt (DI),</li> <li>▪ codieren und decodieren Daten unter Verwendung des Binärsystems (MI),</li> <li>▪ erläutern ein einfaches Transpositionsverfahren als Möglichkeit der Verschlüsselung (DI),</li> <li>▪ vergleichen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (DI).</li> </ul>	<p><a href="https://www.inf-schule.de/information/darstellunginformation">https://www.inf-schule.de/information/darstellunginformation</a>  <a href="https://www.dominikus-gymnasium.de">Codes knacken - St. Dominikus Mädchengymnasium Karlsruhe (dominikus-gymnasium.de)</a>  <a href="https://www.cryptool.org/de/cto/">https://www.cryptool.org/de/cto/</a></p>
<p><b>V. Einführung in die Programmierung (ca. 12 UE)</b></p> <ul style="list-style-type: none"> <li>▪ Was ist ein Algorithmus</li> <li>▪ Einfache Befehlsfolgen und Programmablaufpläne entwerfen und umsetzen</li> <li>▪ Variablenkonzept</li> <li>▪ Fallunterscheidungen entwerfen und umsetzen</li> <li>▪ Schleifen entwerfen und umsetzen</li> <li>▪ Definieren von Befehlsblöcken</li> </ul> <p><b>Programmierprojekt (ca. 6 UE):</b> Entwicklung eines kleinen Spiels in Einzelarbeit oder Kleingruppen</p>	<p><b>Algorithmen</b></p> <p>Algorithmen und algorithmische Grundkonzepte                  Implementation von Algorithmen</p>	<p><b>Algorithmen</b></p> <ul style="list-style-type: none"> <li>▪ formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI),</li> <li>▪ überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI),</li> <li>▪ führen Handlungsvorschriften schrittweise aus (MI),</li> <li>▪ identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI)</li> <li>▪ implementieren Algorithmen in einer visuellen Programmiersprache (MI)</li> <li>▪ implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI),</li> <li>▪ überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI),</li> <li>▪ ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI),</li> </ul>	<p>Als visuelle Programmierumgebung wird <a href="#">Scratch</a> vorgeschlagen.</p> <p>Zur Vorbereitung der Schülerinnen und Schüler auf die anschließende Projektphase wird ein projektbasierter Ansatz gewählt, bei dem die einzelnen Programmierkonzepte stets anhand funktionsfähiger kleiner Spiele erarbeitet werden.</p>



		<ul style="list-style-type: none"> <li>▪ bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A).</li> </ul>	
<p><b>VI. Privatsphäre und Datenschutz (ca. 6 UE)</b></p> <ul style="list-style-type: none"> <li>▪ Personenbezogene Daten / Recht am eigenen Bild</li> <li>▪ Sichere Passwörter</li> <li>▪ Webcams</li> <li>▪ Sichere Nutzung von Messangern und Social-Media</li> <li>▪ ...</li> </ul>	<p><b>Informatik, Mensch und Gesellschaft</b></p> <ul style="list-style-type: none"> <li>▪ Informatiksysteme in der Lebens- und Arbeitswelt</li> <li>▪ Datenbewusstsein</li> <li>▪ Datensicherheit und Sicherheitsregeln</li> </ul>	<p><b>Informatik, Mensch und Gesellschaft</b></p> <ul style="list-style-type: none"> <li>▪ beschreiben anhand von ausgewählten Beispielen die Verarbeitung und Nutzung personenbezogener Daten (DI),</li> <li>▪ erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A),</li> <li>▪ beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A),</li> <li>▪ erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK).</li> </ul>	<p>Da Kinder und Jugendliche die größten Datenmengen über ihr Smartphone produzieren, kann dieses im Unterricht genutzt werden, um potenzielle Mängel unmittelbar zu erkennen.</p>
<p><b>VII. Automaten und künstliche Intelligenz (ca. 7 UE)</b></p> <p><i>Automaten</i></p> <ul style="list-style-type: none"> <li>▪ Zustandsbasierte Automaten aus der Lebenswelt: Getränkeautomat, Fahrkartenautomat etc.</li> <li>▪ Erkennen von Eingabefolgen durch Akzeptoren</li> </ul> <p><i>Künstliche Intelligenz</i></p> <ul style="list-style-type: none"> <li>▪ Entscheidungsbäume</li> <li>▪ Was ist Intelligenz</li> <li>▪ Funktionsweise neuronaler Netze</li> </ul>	<p><b>Automaten und künstliche Intelligenz</b></p> <ul style="list-style-type: none"> <li>▪ Aufbau und Wirkungsweise einfacher Automaten</li> <li>▪ Maschinelles Lernen mit Entscheidungsbäumen</li> <li>▪ Maschinelles Lernen mit neuronalen Netzen</li> </ul>	<p><b>Automaten und künstliche Intelligenz</b></p> <ul style="list-style-type: none"> <li>▪ erläutern die Funktionsweise eines Automaten aus ihrer Lebenswelt (A),</li> <li>▪ stellen Abläufe in Automaten graphisch dar (DI),</li> <li>▪ benennen Anwendungsbeispiele künstlicher Intelligenz aus ihrer Lebenswelt (A),</li> <li>▪ stellen das Grundprinzip eines Entscheidungsbaumes enaktiv als ein Prinzip des maschinellen Lernens dar (DI),</li> <li>▪ beschreiben die grundlegende Funktionsweise künstlicher neuronaler Netze in verschiedenen Anwendungsbeispielen (KK).</li> </ul>	<p>Entscheidungsbäume können gegebenenfalls bereits im Unterrichtsvorhaben V bei der Einführung von Fallunterscheidungen thematisiert werden. Die Grundlagen neuronaler Netze lassen sich in der Erprobungsstufe lediglich spielerisch thematisieren. Es eignet sich bspw. das folgende Material: <a href="http://www.cs4fn.org/machinelearning/trainaneuron.php">http://www.cs4fn.org/machinelearning/trainaneuron.php</a></p>



## Differenzierungskurs Klasse 9 und 10

Unterrichtsvorhaben und konkrete Umsetzung	Inhaltliche Schwerpunkte	Konkretisierte Kompetenzerwartungen	Hinweise zur Umsetzung
<p><b>I. Erfassung, Verarbeitung und Verwaltung von Daten mit Hilfe einer Tabellenkalkulation</b></p> <p>Nutzung einer Tabellenkalkulation:</p> <ul style="list-style-type: none"> <li>▪ <b>Spalten, Zeilen, Zellen</b></li> <li>▪ <b>relative und absolute Adressierung</b></li> <li>▪ <b>einfache Rechenoperationen</b></li> <li>▪ <b>Funktionen</b></li> <li>▪ <b>zusammengesetzte Funktionen</b></li> <li>▪ <b>komplexere Funktionen (beispielsweise zum Suchen von Informationen)</b></li> <li>▪ <b>Sortieren</b></li> <li>▪ <b>Formeln über mehrere Tabellenblätter</b></li> <li>▪ <b>Diagramme</b></li> <li>▪ <b>Modellieren von Problemsituationen als Tabelle</b></li> <li>▪ <b>Pivottabellen</b></li> </ul>	<p><b>Informationen und Daten</b></p> <ul style="list-style-type: none"> <li>▪ Daten und ihre Codierung</li> <li>▪ Erfassung, Verarbeitung und Verwaltung von Daten</li> </ul> <p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>▪ Anwendung von Informatiksystemen</li> </ul>	<p><i>Die Schülerinnen und Schüler...</i></p> <p><b>Informationen und Daten</b></p> <ul style="list-style-type: none"> <li>▪ wählen geeignete Datentypen im Kontext eines Anwendungsbeispiels aus (MI),</li> <li>▪ interpretieren Daten aus dem Ergebnis eines Verarbeitungsprozesses (DI),</li> </ul> <p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>▪ identifizieren für (vernetzte) Informatiksysteme kriteriengeleitet Anwendungsbereiche in der Lebens- und Berufswelt (A),</li> <li>▪ wenden zielgerichtet Prinzipien der strukturierten Dateiverwaltung an (MI).</li> </ul>	<p>Als Tabellenkalkulationsprogramm wird Microsoft Excel verwendet.</p>
<p><b>II. Grundlegender Aufbau von Informatiksystemen durch Rückgriff auf boolesche Algebra und einfache logische Schaltungen</b></p> <ul style="list-style-type: none"> <li>▪ Aussagenlogik</li> <li>▪ <b>Boolesche Algebra / boolesche Funktionen</b></li> <li>▪ Vereinfachen boolescher Funktionen durch KV-Diagramme</li> <li>▪ <b>logische Schaltungen</b></li> <li>▪ <b>Binärsystem, Stellenwertsysteme</b></li> <li>▪ Rechenschaltungen bis hin zum Addierwerk mit Halb- und Volladdierer</li> </ul>	<p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>▪ Logische Schaltungen</li> </ul>	<p><b>Informatiksysteme</b></p> <ul style="list-style-type: none"> <li>▪ erstellen und simulieren logische Schaltungen mithilfe digitaler Werkzeuge (MI),</li> <li>▪ bewerten eine logische Schaltung hinsichtlich ihrer Funktionalität (A).</li> </ul>	<p>Zur Simulation logischer Schaltungen kann das Programm <a href="#">LogikSim</a> oder die Webanwendung <a href="#">CircuitVerse</a> genutzt werden.</p>
<p><b>III. Erstellung und Analyse von Quelltexten mithilfe der Dokumentenbeschreibungssprache HTML</b></p>	<p><b>Automaten und formale Sprachen</b></p>	<p><b>Automaten und formale Sprachen</b></p> <ul style="list-style-type: none"> <li>▪ erläutern die Begriffe Syntax und Semantik einer Programmiersprache an Beispielen (KK),</li> </ul>	<p>Zur Umsetzung kann der Texteditor Notepad++ verwendet werden. Die Anzeige der HTML-Dokumente</p>



<ul style="list-style-type: none"> <li>▪ <b>HTML als Dokumentenbeschreibungssprache</b></li> <li>▪ <b>Syntax und Aufbau von HTML-Dokumenten</b></li> <li>▪ Erstellen einfacher Webauftritte mit Hilfe von HTML</li> </ul>	<ul style="list-style-type: none"> <li>▪ Erstellung und Analyse von Quelltexten</li> </ul>	<ul style="list-style-type: none"> <li>▪ analysieren Quelltexte auf syntaktische Korrektheit (A/MI),</li> <li>▪ erstellen syntaktisch korrekte Quelltexte in einer geeigneten Dokumentenbeschreibungssprache und in einer Programmiersprache (MI).</li> </ul>	<p>erfolgt im vorinstallierten Browser.</p>
<p><b>IV. Codieren und Chiffrieren</b></p> <ul style="list-style-type: none"> <li>▪ Dualsystem</li> <li>▪ ASCII – Code</li> <li>▪ Pixelgrafiken im PBM - Formal</li> <li>▪ Cäsar - Chiffre</li> <li>▪ Kryptoanalyse: Häufigkeitsanalyse und Brute Force</li> <li>▪ Vigenère – Chiffre</li> <li>▪ Kryptoanalyse: Kasiski – Test</li> <li>▪ <i>Optional: Diffie – Hellmann Schlüsselaustausch</i></li> </ul>	<p><b>Daten und ihre Codierung</b></p> <ul style="list-style-type: none"> <li>▪ Daten und ihre Codierung</li> <li>▪ Verschlüsselungsverfahren</li> </ul>	<p><b>Daten und ihre Codierung</b></p> <ul style="list-style-type: none"> <li>▪ beschreiben an ausgewählten Beispielen das Codierungsprinzip von Pixel und Vektorgrafiken (KK),</li> <li>▪ verwenden Substitutionsverfahren als Möglichkeit der Verschlüsselung (MI),</li> <li>▪ beurteilen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (A).</li> </ul>	<p>Die Erarbeitung der Codier- und Chiffriersystemen kann innerhalb eines Escape-Room-Szenarios mit anschließender Debatte über die Vor- und Nachteile der Verfahren insbesondere bei deren Einsatz in Netzwerken erfolgen.</p>
<p><b>V. Informatiksysteme im Detail – Aufbau und gesellschaftlicher Einfluss</b></p> <p>Referate zu Hardware, historischen und gesellschaftlichen Themen mit dazu passenden Plenumsdiskussionen</p> <p><i>Bspw.</i></p> <ul style="list-style-type: none"> <li>▪ Von-Neumann-Architektur</li> <li>▪ Speichertechnologien</li> <li>▪ Entwicklung des PCs</li> <li>▪ Geschichte des Internets</li> <li>▪ Gefahren durch Schadsoftware und Maßnahmen zum Schutz</li> <li>▪ Big Data - Datenschutz am Beispiel der Verarbeitung</li> <li>▪ personenbezogener Daten</li> <li>▪ Ethische Fragestellungen gerade auch im Hinblick auf</li> <li>▪ das christliche Menschenbild im Kontext der künstlichen Intelligenz</li> </ul>	<p><b>Informatik, Mensch und Gesellschaft</b></p> <ul style="list-style-type: none"> <li>▪ <i>Informatiksysteme in der Lebens- und Berufswelt</i></li> <li>▪ <i>Datenschutz und Datensicherheit</i></li> </ul>	<p><b>Informatik, Mensch und Gesellschaft</b></p> <ul style="list-style-type: none"> <li>▪ <i>bewerten verschiedene Lizenzmodelle im Hinblick auf Weiterentwicklung und Nutzung digitaler Produkte (A),</i></li> <li>▪ <i>erläutern die Prinzipien der Datensicherheit (Vertraulichkeit, Integrität und Verfügbarkeit) und berücksichtigen diese beim Umgang mit Daten (A),</i></li> <li>▪ <i>entwickeln kriteriengeleitet Handlungsoptionen für den Umgang mit eigenen und fremden Daten (A),</i></li> <li>▪ <i>diskutieren Auswirkungen des Einsatzes von Informatiksystemen an ausgewählten Beispielen aus der Berufswelt (A/KK).</i></li> </ul>	<p>Anstelle von Referaten kann auch eine Unterrichtsreihe um Thema Informatik, Mensch und Gesellschaft durchgeführt werden.</p> <p>Die angeführten Themen sind lediglich Vorschläge. Bei der Themenfindung können die Schülerinnen und Schüler mit Beratung der Lehrperson eigenständig nach Neigung und Interesse vorgehen.</p>



<ul style="list-style-type: none"> <li>▪ <b>Wandel der Arbeitswelt durch die zunehmende</b></li> <li>▪ <b>Digitalisierung</b></li> </ul>			
<p><b>VI. Aufbau und Wirkungsweise von Automaten</b></p> <ul style="list-style-type: none"> <li>▪ Zustandsübergangsdiagramme und Akzeptoren</li> <li>▪ Überprüfen von Eingabefolgen</li> <li>▪ Entwicklung von Akzeptoren</li> </ul>	<p><b>Automaten und formale Sprachen</b></p> <ul style="list-style-type: none"> <li>▪ Aufbau und Wirkungsweise von Automaten</li> </ul>	<p><b>Automaten und formale Sprachen</b></p> <ul style="list-style-type: none"> <li>▪ analysieren die Funktionsweise eines Automaten mit Hilfe eines Zustandsübergangsdiagramms (DI),</li> <li>▪ entwickeln einen Automaten für eine konkrete Problemstellung (MI).</li> </ul>	<p>Dieses Unterrichtsvorhaben kann je nach inhaltlichem Fortschritt auch ans Ende der Klasse 10 verschoben und inhaltlich gekürzt werden.</p>
<p><b>VII. Einstieg in die Programmierung mit Python</b></p> <p>Erste Programmiererfahrungen mithilfe von Turtlegrafiken</p> <ul style="list-style-type: none"> <li>▪ <b>Verständnis des imperativen Programmierens</b></li> <li>▪ <b>Programmablaufpläne</b></li> <li>▪ <b>einfache Anweisungen</b></li> <li>▪ <b>Variablen</b></li> <li>▪ <b>Parameter</b></li> <li>▪ <b>Methoden</b></li> <li>▪ <b>Verzweigungen</b></li> <li>▪ <b>Schleifen</b></li> <li>▪ Prinzip des modularen Entwurfs</li> </ul>	<p><b>Algorithmen</b></p> <ul style="list-style-type: none"> <li>▪ Algorithmen und algorithmische Grundkonzepte</li> <li>▪ Variablen</li> <li>▪ Implementation von Algorithmen</li> </ul> <p><b>Automaten und formale Sprachen</b></p> <ul style="list-style-type: none"> <li>▪ Erstellung und Analyse von Quelltexten</li> </ul> <p><b>Informationen und Daten</b></p> <p><i>verarbeiten Daten mit einer Programmiersprache unter Berücksichtigung logischer und arithmetischer Operationen (MI).</i></p>	<p><b>Algorithmen</b></p> <ul style="list-style-type: none"> <li>▪ <i>überprüfen algorithmische Eigenschaften (Endlichkeit der Beschreibung, Eindeutigkeit, Terminierung) in Handlungsvorschriften (A),</i></li> <li>▪ <i>stellen Algorithmen in verschiedenen Repräsentationen dar (DI),</i></li> <li>▪ <i>entwerfen und implementieren Algorithmen unter Verwendung von Variablen verschiedener Typen und unter Berücksichtigung des Prinzips der Modularisierung (MI),</i></li> <li>▪ <i>kommentieren, modifizieren und ergänzen Quelltexte von Programmen nach Vorgaben (MI),</i></li> <li>▪ <i>erläutern die Möglichkeit der Werteübergabe mithilfe von Parametern (MI),</i></li> <li>▪ <i>überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen bei der Lösung gleichartiger Probleme (MI),</i></li> <li>▪ <i>beurteilen die Problemangemessenheit verwendeter Algorithmen (MI),</i></li> </ul> <p><b>Automaten und formale Sprachen</b></p> <ul style="list-style-type: none"> <li>▪ <i>erläutern die Begriffe Syntax und Semantik einer Programmiersprache an Beispielen (KK),</i></li> <li>▪ <i>analysieren Quelltexte auf syntaktische Korrektheit (A/MI),</i></li> </ul> <p><b>Informationen und Daten</b></p>	<p>Als Programmiersprache wird Python verwendet. Reduzierte Entwicklungsumgebung mit integrierter Turtlegrafik ist <a href="#">Tigerjython</a>.</p>
<p><b>III. Vertiefung der Programmierung anhand ereignisgesteuerter Programme</b></p> <ul style="list-style-type: none"> <li>▪ <b>Ereigniskonzept in Python</b></li> <li>▪ <b>Mausereignisse</b></li> <li>▪ <b>Tastaturereignisse</b></li> <li>▪ <b>Einbinden eigener Bilder</b></li> <li>▪ <b>Einbinden von Hintergrundgrafiken</b></li> <li>▪ <b>Konzept von Listen in Python</b></li> <li>▪ Ein- und Auslesen von Daten</li> </ul>			



<p><i>Abschluss dieses längeren Unterrichtsvorhabens kann die Entwicklung und Präsentation eines Spiels in (Klein)gruppen sein.</i></p>		<ul style="list-style-type: none"> <li>▪ <i>verarbeiten Daten mit einer Programmiersprache unter Berücksichtigung logischer und arithmetischer Operationen (MI).</i></li> </ul>	
<p><b>IX. Künstliche Intelligenz und maschinelles Lernen</b></p> <ul style="list-style-type: none"> <li>▪ Entscheidungs bäume</li> <li>▪ Was ist Intelligenz</li> <li>▪ Maschinelles Lernen als Teilgebiet der künstlichen Intelligenz</li> <li>▪ Maschinelles Lernen             <ul style="list-style-type: none"> <li>• überwachtes Lernen</li> <li>• bestärkendes Lernen</li> <li>• unüberwachtes Lernen</li> </ul> </li> <li>▪ <i>Optional: K-Means Clusteralgorithmus als Beispiel für unüberwachtes Lernen</i></li> </ul>	<p><b><i>Künstliche Intelligenz und maschinelles Lernen</i></b></p> <ul style="list-style-type: none"> <li>▪ überwachtes Lernen</li> <li>▪ unüberwachtes Lernen</li> <li>▪ bestärkendes Lernen</li> </ul>	<p><b><i>Künstliche Intelligenz und maschinelles Lernen</i></b></p> <ul style="list-style-type: none"> <li>▪ beschreiben Anwendungsbeispiele künstlicher Intelligenz zum überwachtem, unüberwachtem und bestärkenden Lernen (KK),</li> <li>▪ beschreiben die grundlegenden Funktionsweisen maschinellen Lernens (überwacht, unüberwacht, bestärkend) in verschiedenen Anwendungsbeispielen (KK),</li> <li>▪ ordnen begründet die Methoden des maschinellen Lernens (überwachtes Lernen, unüberwachtes, bestärkendes Lernen) verschiedenen Anwendungsbeispielen zu (A),</li> <li>▪ analysieren den Einfluss von Trainingsdaten auf die Ergebnisse eines Verfahrens maschinellen Lernens (A).</li> </ul>	<p>Die einzelnen Klassen maschinellen Lernens werden immer an konkreten Anwendungsbeispielen eingeführt. Als Differenzierungsmöglichkeit können einzelne Verfahren konkret erläutert oder sogar implementiert werden.</p>

## Grundsätze der Leistungsbewertung

Die rechtlich verbindlichen Grundsätze der Leistungsbewertung regelt das Schulgesetz ([§ 48 Grundsätze der Leistungsbewertung](#)) sowie die Ausbildungs- und Prüfungsordnung für die Sekundarstufe I ([§ 6 APO-SI Leistungsbewertung, Klassenarbeiten, Nachteilsausgleich](#)). Ferner beruht die Leistungsbewertung auf dem Kernlehrplan für das Fach Informatik.

Grundsätzlich kann die Leistungsbewertung auf den Ergebnissen *schriftlicher Arbeiten* und der Beurteilung der *sonstigen Mitarbeit* fußen. Zu letzterer gehören alle im Zusammenhang mit dem Unterricht erbrachten mündlichen und praktischen Leistungen gegebenenfalls kurze, schriftliche Übungen. Beide Beurteilungsbereiche werden bei der Notenfindung angemessen berücksichtigt. Um dem unterschiedlichen Entwicklungsgrad der Schülerinnen und Schüler sowie der Vorbereitung auf das Abitur Rechnung zu tragen, unterscheiden sich die Grundsätze der Leistungsbewertung je nach Jahrgangsstufe.

### Klasse 6

Die Leistungsbewertung in der Erprobungsstufe erfolgt ausschließlich auf Grundlage des Beurteilungsbereiches der *sonstigen Leistungen*. Gegenstand dieser sind:

1. Individuelle Beteiligung am Unterrichtsgeschehen
  - Regelmäßigkeit / Kontinuität
  - Qualität / Bedeutung der Beiträge für den Fortschritt des Unterrichts
  - Quantität
2. Einbringen in Gruppenprozesse (Anteil an Partner- oder Gruppenleistungen, Kooperationsfähigkeit)
3. Praktische Arbeit am Computer (Selbstständigkeit, gezielte Nutzung, Zusammenarbeit mit Mitschülerinnen und Mitschülern)
4. Leistungsnachweise in Form von abgeschlossenen Ergebnissen
5. Zusätzliche Einzelleistungen wie Referate

Zu den in Punkt 4 angeführten Leistungsnachweisen zählen unter anderem die Ergebnisse schriftlicher Tests, einzureichende Ergebnisse von Aufgabenbearbeitungen und abgeschlossene Softwareprodukte wie das Ergebnis von Projektarbeiten. Diese Leistungsnachweise können von der Lehrperson unter Berücksichtigung der individuellen Situation in der Lerngruppe im Laufe eines Schuljahres gegebenenfalls mehrfach eingefordert werden. Einzig die Bewertung eines Programmierprojektes ist fest in den Unterrichtsvorhaben verankert.

### Differenzierungskurs Klasse 9 und 19

Das Wahlpflichtfach ist Teil der Fächergruppe II, in der pro Halbjahr schriftliche Klassenarbeiten zu schreiben sind. Entsprechend setzt sich die Gesamtleistung aus der *schriftlichen Leistung* und der *sonstigen Mitarbeit* zusammen.

#### *Schriftliche Arbeiten*

Aufgrund der Aufteilung auf die verschiedenen Themenbereiche ergeben sich die folgenden Möglichkeiten zur Leistungsüberprüfung mittels schriftlicher Arbeiten. Laut Fachkonferenzbeschluss soll im zweijährigen Kurs eine Arbeit durch eine andere Form der Leistungsüberprüfung ersetzt werden (möglich nach APO S1 §6.8), da sie der vielfach in Projekten angelegten Arbeitsweise des Faches entspricht.

#### Jahrgangsstufe 9

- Eine Arbeit zum Thema Tabellenkalkulationen, die am PC geschrieben wird.
- Eine schriftliche Arbeit zum Thema boolesche Algebra und logische Schaltungen.



- Eine Arbeit zum Thema HTML, die am PC geschrieben wird.
- Eine schriftliche Arbeit zum Thema Codieren / Chiffrieren und Automaten

### Klasse 10

- eine Arbeit am PC zum Einstieg in die Programmierung mit Python
- eine schriftliche Arbeit zur Analyse und Funktionsweise von Programmen und Quellcode
- eine Arbeit am PC zu ereignisgesteuerten Programmen
- eine Projektarbeit

In den schriftlichen Arbeiten und den Arbeiten am PC sind die folgenden Anforderungsbereiche in etwa in der angegebenen Gewichtung zu berücksichtigen:

- Reproduktion von erlerntem Wissen, bzw. einfacher im Unterricht behandelter Algorithmen (30%)
- Anwendung des Erlernten unter Berücksichtigung der im Unterricht vermittelten Kompetenzen (60%)
- Transfer sowie Lösen komplexer Probleme und Wertung (10%)

Bei der Benotung sollte folgender Schlüssel zu Grunde gelegt werden:

Note	Leistung
Sehr gut	ab 87 % der erreichbaren Punkte
Gut	ab 74 % der erreichbaren Punkte
Befriedigend	ab 61 % der erreichbaren Punkte
Ausreichend	ab 48 % der erreichbaren Punkte
Mangelhaft	ab 24 % der erreichbaren Punkte
Ungenügend	Darunter

### Sonstige Mitarbeit

Je nach zeitlichem Verlauf können die schriftlichen Arbeiten um weitere schriftliche Überprüfungen (beispielsweise ein Test zum Thema der künstlichen Intelligenz) erweitert werden.

Ferner sind Gegenstand der sonstigen Mitarbeit:

- individuelle Beteiligung am Unterrichtsgespräch (Regelmäßigkeit, Qualität, Bedeutung der Beiträge für den Fortschritt des Unterrichts)
- Einbringen in Gruppenprozesse (Anteile an Gruppenleistungen, Kooperationsfähigkeit)
- praktische Arbeit am Computer (Selbstständigkeit, gezieltes Nutzen von Hilfestellungen durch den Rechner, Zusammenarbeit mit der Mitschülerin oder dem Mitschüler)
- zusätzliche Einzelleistungen wie Referate

Bei der Notenbildung sind die schriftlichen Arbeiten und der Bereich "Sonstige Mitarbeit" angemessen zu berücksichtigen. Dabei sollte der Blick auch auf die individuelle Entwicklung der jeweiligen Schülerin oder des jeweiligen Schülers gelenkt und diese berücksichtigt werden. Eine Rückmeldung zur Sonstigen Mitarbeit zum Quartal in Form einer Note ist im Fach Informatik aufgrund der Struktur des Unterrichts (lange Übungsphasen, Einsatz des PCs) nicht nach jedem Quartalsende angemessen.

## Individuelle Förderung und Differenzierungsmaßnahmen



Im Fach Informatik geht – insbesondere bei der praktischen Arbeit am Computer – die Schere der Leistungsfähigkeit schnell auseinander. Dies kann unterschiedliche Ursachen haben, denen durch verschiedene Ansätze zur Differenzierung und individuellen Förderung Rechnung getragen wird.

### ***Differenzierung nach Lerntempo***

Immer wieder kommt es vor, dass einzelne Schülerinnen und Schüler die Aufgabenstellungen bedeutend schneller als ihre Mitschülerinnen und Mitschüler bearbeitet haben. Dies kann im Unterricht dadurch aufgefangen werden, dass diese Schülerinnen und Schüler den übrigen für den Rest der Arbeitsphase als ‚Experten‘ Hilfestellungen geben.

Eine weitere Möglichkeit ist der Einsatz von ‚Experten-‘ oder ‚Sprinteraufgaben‘, die über den verpflichtenden Unterrichtsstoff hinausgehen und auch leistungsstarke Schülerinnen und Schüler vor Herausforderungen stellen, bei denen sie ihr informatisches Wissen vertiefen können.

### ***Differenzierung durch offene Aufgabenstellungen***

Häufig tauchen im Informatikunterricht Aufgabenstellungen auf, bei denen der Lösungsweg oder sogar das konkrete Lernprodukt zunächst offen ist. Dies ist insbesondere im Bereich der Programmierung – besonders bei Programmierprojekten - der Fall, wo in der Regel kein eindeutiger Lösungsweg fixierbar ist. Hier können die Schülerinnen und Schüler auf Grundlage ihrer individuellen Kompetenzen und Vorerfahrungen selbstständig über Umfang und Komplexität ihrer Lernprodukte entscheiden und stellen dadurch eine Form der natürlichen Differenzierung her.

### ***Förderung außerhalb des Unterrichts***

Es ist nicht ungewöhnlich, dass einzelne Schülerinnen und Schüler ein großes Interesse an informatischen Themen entwickeln, dass schnell über den im Unterricht behandelten Stoff hinausgeht. In diesem Fall besteht die Möglichkeit, an Angeboten außerhalb des regulären Unterrichts teilzunehmen. So werden jährlich verschiedene Informatikwettbewerbe wie der [Informatik-Biber](#) oder der [Jugendwettbewerb Informatik](#) statt ausgetragen, an denen eine Teilnahme prinzipiell immer möglich ist.

Ferner besteht die Möglichkeit, an Angeboten außerhalb der der Schule wie beispielsweise der [Junioruni Essen](#) teilzunehmen, wo regelmäßig Kurse zu verschiedenen Inhalten aus dem Bereich der Informatik angeboten werden.

### ***Förderung von Mädchen in der Informatik***

Zahlreiche Studien zeigen, dass mit steigendem Alter der Anteil der Mädchen in Informatikkursen stetig abnimmt. Diesem Trend soll am B. M. V. – Gymnasium aktiv entgegengewirkt werden. Im Informatikunterricht wird aus diesem Grund verstärkt auf die Förderung von Mädchen geachtet. Dies kann durch verschiedene Maßnahmen erreicht werden:

- Einsatz von lebensweltnahen Szenarien und Lernaktivitäten
- Vermeiden von Stereotypen
- Aufzeigen weiblicher Vorbilder aus dem Bereich der Informatik
- Aufzeigen von interdisziplinären und sozialen Zusammenhängen des Fachs

Darüber hinaus gibt es verschiedene Angebote aus dem Bereich der Informatik, die sich speziell auf die Förderung von Mädchen beschränken. Beispielweise [girls@BWINE](#) oder die [GIRLS Hacker School](#).



## Literatur

- [1] Happe, Lucia; Buhnova, Barbora, Koziolk, Anne und Wagner, Ingo (2021): Effective measures to foster girls' interest in secondary computer science education, In: *Educ Inf Technol* Vol. 26, Springer Verlag, (S. 2811–2829)
- [2] Hrsg.: Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen (2021): Kernlehrplan für die Sekundarstufe I - Klasse 5 und 6 in Nordrhein-Westfalen – Informatik, Düsseldorf
- [3] Hrsg.: Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen (2023): Kernlehrplan für die Sekundarstufe I – Wahlpflichtfach Informatik – Informatik, Düsseldorf
- [4] Poloczek, Jürgen (2014): Zur Konstruktion von Aufgaben. Aufgabentypisierung und das Konzept der Blütenaufgaben. In: LOG IN: Vol. 34, No. 1., Berlin: LOG IN Verlag. (S. 33 – 39)
- [5] Schubert, Sigrid und Schwill, Andreas (2011): Didaktik der Informatik, 2. Auflage; Heidelberg: Spektrum Akademischer Verlag

**B.M.V.-Gymnasium Essen  
Schulinterner Lehrplan  
Sekundarstufe II**

**Informatik**

**(Stand: 04.11.2015)**

# Inhalt

<b>1 Die Fachgruppe Informatik des B.M.V.-Gymnasiums Essen</b>	<b>3</b>
<b>2 Entscheidungen zum Unterricht</b>	<b>4</b>
2.1 Unterrichtsvorhaben	4
2.1.1 Übersichtsraster Unterrichtsvorhaben	5
I) Einführungsphase	5
II) Qualifikationsphase	10
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit	21
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung	22
<b>3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen</b>	<b>24</b>
<i>Zusammenarbeit mit anderen Fächern</i>	24
<i>Fächerverbindendes Projekt der Q1 und Projektstage</i>	24
<i>Vorbereitung auf die Erstellung der Facharbeit</i>	24
<i>Exkursionen</i>	24
<b>4 Qualitätssicherung und Evaluation</b>	<b>25</b>

## **Die Fachgruppe Informatik des B.M.V.-Gymnasiums Essen**

Beim B.M.V.- Gymnasium handelt es sich um eine fünf bis sechszügige Schule im Essener Stadtteil Holsterhausen mit zurzeit ca. 1200 Schülerinnen und Schülern.

Das Fach Informatik wird am B.M.V.-Gymnasium ab der Jahrgangsstufe 8 im Wahlpflichtbereich II (WP II) dreistündig unterrichtet. In der zweijährigen Laufzeit dieser Kurse wird in altersstufengerechter Weise unter anderem auf Grundlagen der Programmierung mit Hilfe von Turtle-Grafiken in der Programmiersprache Python, auf die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen, auf Robotik und auf die Benutzung von Standard-Office-Software eingegangen.

In der Sekundarstufe II bietet das B.M.V.-Gymnasium in EF meist zwei Grundkurse, in Q1 und Q2 je nach Wahlverhalten je einen Leistungskurs und ein bis zwei Grundkurse an.

Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik des B.M.V.-Gymnasiums aus vier Lehrkräften, denen zwei Computerräume mit jeweils 31 Computerarbeitsplätzen zur Verfügung stehen.

Der Unterricht erfolgt im 67,5-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse zwei Langstunden und für Leistungskurse drei ein Drittel Langstunden vor.

## 2 Entscheidungen zum Unterricht

### 2.1 Unterrichtsvorhaben

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 in Form eines Übersichtsrasters, in welchem Konkretisierungen bereits eingearbeitet wurden.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, 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, aktuelle 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.

---

## 2.1.1 Übersichtsraster Unterrichtsvorhaben

### I) Einführungsphase

#### Unterrichtsvorhaben E-I

**Thema:**

*Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten*

**Kompetenzen:**

Die Schülerinnen und Schüler

- beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A),
- nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Konkretisierung**

Im Rahmen dieser Unterrichtssequenz lernen die SuS den Umgang mit den schuleigenen Informatiksystemen kennen. Diese Sequenz dient vor allem dazu unterschiedliche Niveaus im Umgang mit der Hardware anzugleichen und den Kurs auf einen einheitlichen Level zu führen, um eine Arbeitsgrundlage für die folgenden Unterrichtssequenzen zu haben.

Inhaltlich sollten die SuS folgende Bereiche beherrschen:

- Umgang mit dem Windows-Rechner
- Starten von Programmen
- Einrichten von Ordnern
- Speichern, Öffnen und sinnvolles Benennen von Dateien
- Benennen von Hardwarekomponenten und Einordnung in ein von-Neumann-Modell

**Zeitbedarf:**

2 Stunden

---

## Unterrichtsvorhaben E-II

### **Thema:**

*Grundlagen der Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Konsolenprogrammen*

### **Kompetenzen:**

Die Schülerinnen und Schüler

- analysieren und erläutern einfache Algorithmen und Programme (A),
- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),
- modifizieren einfache Algorithmen und Programme (I),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- testen Programme schrittweise anhand von Beispielen (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

### **Inhaltsfelder:**

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten

### **Konkretisierung**

Anhand des Scripts „Einführung in die Programmierung mit Java“ der RWTH Aachen lernen die SuS Syntax, Kontrollstrukturen und Variablenkonzept der Programmiersprache Java kennen. Konkret beherrschen sie:

- den Umgang mit dem Java-Editor
- Aufbau eines Java-Programms
- Datentypen und –operationen
- Variablen
- Verzweigungen
- Schleifen
- Darstellung von Programmabläufen in

**Zeitbedarf:** 12 Stunden

---

## Unterrichtsvorhaben E-III

### **Thema:**

*Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand von statischen Grafikszenen*

### **Kompetenzen:**

Die Schülerinnen und Schüler

- ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),
- stellen die Kommunikation zwischen Objekten grafisch dar (M),
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- stellen den Zustand eines Objekts dar (D).
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),
- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),
- modifizieren einfache Algorithmen und Programme (I),
- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),
- implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),
- implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),
- testen Programme schrittweise anhand von Beispielen (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I).

### **Inhaltsfelder:**

- Daten und ihre Strukturierung
- Formale Sprachen und Automaten

### **Konkretisierung:**

Anhand einfacher Beispiele (z.B. Eisenbahn, Orchester) identifizieren die SuS die beteiligten Objekte und die für sie relevanten Methoden und Attribute, stellen diese in Implementationsdiagrammen dar und implementieren die entsprechenden Klassen.

**Zeitbedarf:** 16 Stunden

---

## Unterrichtsvorhaben E-IV

### **Thema:**

*Modellierung und Implementierung von Klassen- und Objektbeziehungen*

### **Kompetenzen:**

Die Schülerinnen und Schüler

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

### **Konkretisierung:**

Aufbauend auf den Szenarien des Unterrichtsvorhabens E-III erkennen die SuS Beziehungen zwischen den beteiligten Klassen, benennen diese, stellen sie im Implementationsdiagramm dar und implementieren die Beziehungen.

**Zeitbedarf:** 12 Stunden

---

## Unterrichtsvorhaben E-V

### **Thema:**

*Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes*

### **Kompetenzen:**

Die Schülerinnen und Schüler

- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),
- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A),
- stellen ganze Zahlen und Zeichen in Binärcodes dar (D),
- interpretieren Binärcodes als Zahlen und Zeichen (D),
- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).

### **Inhaltsfelder:**

- Informatik, Mensch und Gesellschaft
- Informatiksysteme

### **Konkretisierung:**

Anhand von Schülervorträgen und Filmen

- zur Geschichte der Datenverarbeitung
- zum Thema Datenschutz
- zur Darstellung von Zahlen und Zeichen in Binärcode
- zur Rolle von großen IT-Unternehmen in der Gesellschaft
- zu aktuellen gesellschaftlichen Fragestellungen

Eine Exkursion in das Heinz-Nixdorf-Museumsforum in Paderborn würde sich im Rahmen dieser Reihe anbieten.

**Zeitbedarf:** 8 Stunden

---

## II) Qualifikationsphase

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).

---

## Unterrichtsvorhaben Q1-I:

**Thema:** Wiederholung der objektorientierten Modellierung und Programmierung mit dem Schwerpunkt Vererbung

**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? Wie funktioniert das Konzept der Vererbung?*

### Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern objektorientierte Modellierungen (A),
- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),
- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),
- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),
- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),
- 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),
- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),
- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),
- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),
- dokumentieren Klassen (D).

### 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. Im Grundkurs werden mehr Teile vorgegeben als im Leistungskurs.

Die Schülerinnen und Schüler 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.

**Zeitbedarf:** 6 Stunden (LK 8 Stunden)

---

## Unterrichtsvorhaben Q1-II:

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

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

### Kompetenzen

Die Schülerinnen und Schüler

- 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),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- modifizieren Algorithmen und Programme (I),
- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (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 lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).

### 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. Die Klasse Queue wird dabei im Grundkurs von der Lehrkraft vorgegeben, im LK wird diese gemeinsam entwickelt. Anschließend wird in einem kleinen Projekt der Umgang mit der Datenstruktur geübt. 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. Im LK wird die Klasse Stack selbstständig implementiert.

**Zeitbedarf:** 13 Stunden (LK 20 Stunden)

---

## **Unterrichtsvorhaben Q1-III:**

**Thema:** Suchen und Sortieren auf linearen Datenstrukturen

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

### **Kompetenzen**

Die Schülerinnen und Schüler

- 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),
- 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).

### **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 Implementationen von Quicksort sowie dem Sortieren durch Einfügen werden analysiert und erläutert, im LK auch implementiert. Die rekursive Abarbeitung eines Methodenaufrufs von Quicksort wird grafisch dargestellt.

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

**Zeitbedarf:** 10 Stunden (LK 16 Stunden)

---

## 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?*

### Kompetenzen

Die Schülerinnen und Schüler

- 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),
- bestimmen Primär- und Sekundärschlüssel (M),
- überführen Datenbankschemata in vorgegebene Normalformen (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),
- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).

### 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. Als Material können hier Ausschnitte aus dem Buch Informatik 2 (Klett) oder Informatik 2 (Schöningh) verwendet werden.

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. Dabei sind die im LK verwendeten SQL-Anweisungen deutlich komplexer als im Grundkurs.

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.

**Zeitbedarf:** 13 Stunden (LK 20 Stunden)

## **Unterrichtsvorhaben Q1-V: (nur Leistungskurs)**

**Thema:** Netzstrukturen

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

### **Kompetenzen:**

Die Schülerinnen und Schüler

- beschreiben und erläutern 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 die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),
- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A),
- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).
- analysieren und erläutern Protokolle zur Kommunikation in einem Client-ServerNetzwerk (A),
- entwickeln und erweitern Protokolle zur Kommunikation in einem Client-ServerNetzwerk (M).
- analysieren und erläutern Algorithmen und Methoden zur Client-Server-Kommunikation (A),
- entwickeln und implementieren Algorithmen und Methoden zur Client-ServerKommunikation (I).

---

### **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. Der Umgang mit den Zentralabiturklassen Client und Server werden erlernt und anhand eines eigenen kleinen Projekts (z.B. Chat) umgesetzt.

**Zeitbedarf:** 10 Stunden

### **Unterrichtsvorhaben Q2-I:**

**Thema:** Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen (Bäume)

**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?*

### **Kompetenzen:**

- 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),
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),
- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),
- 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),
- modifizieren Algorithmen und Programme (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 lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),
  - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).

### **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. Im LK werden die Baummethoden selbst implementiert.

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  $\leftrightarrow$  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.

Im LK wird mindestens die Methode `search` des `BinarySearchTree` implementiert.

**Zeitbedarf:** 16 Stunden (24 Stunden)

---

## Unterrichtsvorhaben Q2-II:

**Thema:** Endliche Automaten und formale Sprachen

### Kompetenzen

Die Schülerinnen und Schüler

- analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A),
- analysieren und erläutern Grammatiken regulärer Sprachen (A),
- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A),
- ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),
- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M),
- entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M),
- entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M),
- modifizieren Grammatiken regulärer Sprachen (M),
- entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M),
- stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D),
- ermitteln die Sprache, die ein endlicher Automat akzeptiert (D).
- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).

**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.

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.

Im Leistungskurs wird zusätzlich ein Automat in einen Parser überführt. Weiter wird an einem Beispiel ein Kellerautomat untersucht und mit Hilfe der Chomsky-Hierarchie der Unterschied zwischen regulären und kontextfreien Sprachen und Grammatiken behandelt.

**Zeitbedarf:** 14 Stunden (LK 20 Stunden)

## **Unterrichtsvorhaben Q2-III: (nur LK)**

**Thema:** Graphen als weiteres Beispiel für nicht lineare Datenstrukturen

**Leitfragen:** *Welche Darstellungsformen von Graphen gibt es? Welche Anwendungssituationen lassen sich mit Hilfe von Graphen veranschaulichen? Wie funktioniert das Berechnen kürzester Wege?*

### **Kompetenzen:**

- 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),
- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (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),
- modifizieren Algorithmen und Programme (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 lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),
- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).

---

**Vorhabenbezogene Konkretisierung:**

Nach einem kurzen Exkurs in verschiedene Darstellungsformen von Graphen (Adjazenz- und Inzidenzmatrizen) werden mit und ohne Hilfe der Zentralabiturklasse Graph folgende Problemstellungen bearbeitet:

- Eulerwege und -kreise
- Breiten- und Tiefensuche
- minimale Spannbäume (Algorithmen von Kruskal und Prim)
- Finden kürzester Wege mit Hilfe des Dijkstra-Algorithmus

Darüber hinaus werden kleiner Anwendungen mit Hilfe der Klasse Graph umgesetzt.

**Zeitbedarf:** 12 Stunden

**Unterrichtsvorhaben Q2-IV:**

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

---

## 2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des B.M.V.-Gymnasiums die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 14 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 15 bis 21 sind fachspezifisch angelegt.

### Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schüler/innen erreichen einen Lernzuwachs.
- 6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
- 7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
- 9) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- 10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- 11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- 12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- 13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 14) Es herrscht ein positives pädagogisches Klima im Unterricht.

### Fachliche Grundsätze:

- 15) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 16) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- 17) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 18) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- 19) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 20) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 21) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

## 2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

### Klausuren in der Sekundarstufe 2

#### Anzahl und Dauer (in 45-Minuten-Einheiten)

	EF 1.HJ		EF 2.HJ		Q1.1		Q1.2		Q2.1		Q2.2	
	Anzahl	Dauer	Anzahl	Dauer	Anzahl	Dauer	Anzahl	Dauer	Anzahl	Dauer	Anzahl	Dauer
LK					2	3	2	3	2	4	1	4,25h
GK	1	2	1	2	2	2	2	2	2	3	1	3h

In den schriftlichen Arbeiten sind die folgenden Anforderungsbereiche in etwa in der angegebenen Gewichtung zu berücksichtigen:

- I) Reproduktion von erlerntem Wissen (30%)
- II) Anwendung des Erlernten unter Berücksichtigung der im Unterricht vermittelten Kompetenzen (60%)
- III) Transfer sowie Lösen komplexer Probleme und Wertung (10%)

Bei der Benotung sollte folgender Schlüssel zu Grunde gelegt werden:

Note	1+	1	1-	2+	2	2-	3+	3	3-	4+	4	4-	5+	5	5-	6
Punkte	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ab (%)	95	90	85	80	75	70	65	60	55	50	45	39	33	27	20	0

---

## **Sonstige Mitarbeit in der Sekundarstufe II**

Gegenstand der Sonstigen Mitarbeit in der Sekundarstufe II sind:

- individuelle Beteiligung am Unterrichtsgespräch (Regelmäßigkeit, Qualität, Bedeutung der Beiträge für den Fortschritt des Unterrichts)
- Einbringen in Gruppenprozesse (Anteile an Gruppenleistungen, Kooperationsfähigkeit)
- praktische Arbeit am Computer (Selbstständigkeit, gezieltes Nutzen von Hilfestellungen durch den Rechner, Zusammenarbeit mit der Mitschülerin oder dem Mitschüler)
- zusätzliche Einzelleistungen wie Referate

## **Halbjahresnote**

Den Schülerinnen und Schülern wird am Ende jedes Quartals die Note für den Bereich "Sonstige Mitarbeit" mitgeteilt. Bei der Bildung der Halbjahresnote sind die schriftlichen Arbeiten und der Bereich "Sonstige Mitarbeit" zu gleichen Teilen zu berücksichtigen.

---

## **3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen**

### **Zusammenarbeit mit anderen Fächern**

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden.

### **Fächerverbindendes Projekt der Q1 und Projekttag**

Jährlich werden am B.M.V.-Gymnasium fächerübergreifende Projekte für die Jahrgangsstufe Q1 und in unregelmäßigen Abständen Projekttag für 5 bis EF angeboten. Die Fachkonferenz Informatik bietet in diesem Zusammenhang jeweils mindestens ein Projekt an.

### **Vorbereitung auf die Erstellung der Facharbeit**

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

### **Exkursionen**

In der Einführungsphase wird im Rahmen des Unterrichtsvorhabens „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum Heinz Nixdorf MuseumsForum durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

Für die Jahrgangsstufe Q1 hat sich in den letzten Jahren eine Fahrt zur langen Nacht der Wissenschaften in Berlin etabliert. Neben dem Besuch der Wissenschaftsnacht sollen hier informatische Themen beispielsweise beim Besuch des Technikmuseums, des Kommunikationsmuseum und des Computerspielmuseums im Vordergrund stehen.

---

## 4 Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Erstmals nach Ende der Einführungsphase im Sommer 2015, werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Einführungsphase beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.