

**Städt. Gymnasium Dionysianum Rheine:**

**Schulinterner Lehrplan zum Kernlehrplan für die gymnasiale Oberstufe (ab Punkt 2 identisch mit dem des kooperierenden Emsland-Gymnasiums)**

## **Informatik**

**(verabschiedet FK: 24.06.2015, redaktionell bearbeitet bzgl. MKR und Verbraucherbildung in 2019.05)**

<b>1. RAHMENBEDINGUNGEN DER FACHLICHEN ARBEIT</b>	<b>4</b>
1.1. Lage der Schule	4
1.2. Aufgaben des Fachs bzw. der Fachgruppe in der Schule vor dem Hintergrund der Schülerschaft	4
1.3. Funktionen und Aufgaben der Fachgruppe vor dem Hintergrund des Schulprogramms	4
1.4. Verfügbare Ressourcen	6
1.5. Funktionsinhaber/innen der Fachgruppe	6
<b>2. ENTSCHEIDUNGEN ZUM UNTERRICHT</b>	<b>7</b>
2.1. Unterrichtsvorhaben	7
2.2. Übersichtsraster Unterrichtsvorhaben	8
2.2.1. Einführungsphase	8
2.2.2. Qualifikationsphase (Q1 und Q2)	10
2.3. Konkretisierte Unterrichtsvorhaben	12
2.3.1. Einführungsphase	12
2.3.1.1. Unterrichtsvorhaben EF-I	13
2.3.1.2. Unterrichtsvorhaben EF-II	14
2.3.1.3. Unterrichtsvorhaben EF-III	16
2.3.1.4. Unterrichtsvorhaben EF-IV	17
2.3.1.5. Unterrichtsvorhaben EF-V	18
2.3.1.6. Unterrichtsvorhaben EF-VI	20
2.3.1.7. Unterrichtsvorhaben EF-VII	22
2.3.2. Qualifikationsphase	23
2.3.2.1. Unterrichtsvorhaben Q1-I	23
2.3.2.2. Unterrichtsvorhaben Q1-2	25
2.3.2.3. Unterrichtsvorhaben Q1-III	29
2.3.2.4. Unterrichtsvorhaben Q1-IV	31
2.3.2.5. Unterrichtsvorhaben Q2-I	33
2.3.2.6. Unterrichtsvorhaben Q2-II	36
2.3.2.7. Unterrichtsvorhaben Q2-III	38
2.3.2.8. Unterrichtsvorhaben Q2-IV	41
<b>3. GRUNDSÄTZE DER FACHMETHODISCHEN UND FACHDIDAKTISCHEN ARBEIT</b>	<b>42</b>
3.1. Grundsätze der Leistungsbewertung und Leistungsrückmeldung	43
3.1.1. Beurteilungsbereich Klausuren	43
3.1.2. Beurteilungsbereich Sonstige Mitarbeit	43
3.1.2.1. Verbindliche Absprachen der Fachkonferenz	44
3.1.2.2. Leistungsaspekte	44
3.1.2.3. Kriterien	44
3.1.2.4. Grundsätze der Leistungsrückmeldung und Beratung	45
<b>4. ENTSCHEIDUNGEN ZU FACH- UND UNTERRICHTSÜBERGREIFENDEN FRAGEN</b>	<b>45</b>
4.1. Zusammenarbeit mit anderen Fächern	45
4.2. Vorbereitung auf die Erstellung der Facharbeit	45
<b>5. QUALITÄTSSICHERUNG UND EVALUATION</b>	<b>46</b>

*Ein schulinterner Lehrplan ist wie folgt gestaltet:*

- 1. Er ordnet den Jahrgangsstufen obligatorische Unterrichtsvorhaben zu (schulbezogene Schwerpunktsetzungen).*
- 2. Der Lehrplan enthält zu jedem Unterrichtsvorhaben folgende Angaben:  
Kompetenzerwartungen des Kernlehrplans, zu denen das Vorhaben einen Beitrag leistet  
Thema und inhaltliche Schwerpunkte  
Bezüge zu fachlichen Rahmenbedingungen und curricular relevanten Aspekten des Schulprogramms  
Hinweise zur Gestaltung der individuellen Förderung ggf. Inklusion.*
- 3. Der Lehrplan weist grundlegende Entscheidungen bzw. Vereinbarungen aus zu  
Verfahren und Kriterien der Leistungsbewertung  
didaktisch-methodischen Prinzipien der Unterrichtsgestaltung  
fachübergreifenden Kooperationen (ggf. Jahrgangspartituren)  
Lernmitteln  
Verknüpfungen des Unterrichts mit Ganztagsangeboten (z.B. Projekten).  
der Einbeziehung außerschulischer Partner in den Unterricht  
der Nutzung außerschulischer Lernorte für den Unterricht  
Evaluationsmaßnahmen und Auswertungsverfahren*

Unsere schulinternen Lehrpläne orientieren sich an den Vorgaben der jeweiligen Kernlehrpläne unter besonderer Berücksichtigung der Prüfkriterien des MSW: Verknüpft wird dies mit dem Schulprogramm, dem Methoden- und Mediencurriculum des Dionysianum sowie den Verfahren und Kriterien der Leistungsbewertung.

# **1. Rahmenbedingungen der fachlichen Arbeit**

## **1.1. Lage der Schule**

Seit mehr als 360 Jahren ist unsere Schule ein Haus des Lernens zwischen Tradition und Aufbruch.

Wir, die Schulgemeinde des Dionysianum, wollen zusammen arbeiten und zusammen leben. Unsere Schule ist ein Haus des Lernens, das Schülerinnen und Schüler, Lehrerinnen und Lehrerinnen sowie die Eltern in geteilter Verantwortung prägen und gestalten.

Unser Leitziel ist, dass unsere Schülerinnen und Schüler eigenständige Persönlichkeiten werden, die ihrer Verantwortung für sich selbst und für die Gesellschaft gerecht werden und mit dem erfolgreichen Abschluss am Gymnasium Dionysianum die Grundlage für lebenslanges Lernen erlangt haben.

Das Dionysianum steht dabei in Konkurrenz mit den anderen Gymnasien nicht nur um Rheiner Schülerinnen und Schüler, sondern auch um Schüler\*Innen aus dem nahen Niedersachsen mit den Gymnasien in Lingen, Bentheim und Handrup, welche zum Schuljahr 2015/16 auf G9 umstellen. Dieser Konkurrenz kann nur mit einem überzeugenden Programm über Unterricht hinaus (IP-Klassen, Soziales Lernen (Lionsquest), Komm-mit bzw. Zukunftsschule, Englisch intensiv, NT-Unterricht, Schule der Zukunft, Schule gegen Rassismus), Transparenz und Elternarbeit nebst Förderverein und Ehemaligenarbeit im Verein alter Dionysianer VAD begegnet werden.

Das Dionysianum gehört zum Standorttyp 2 und ist verkehrstechnisch günstig gelegen.

## **1.2. Aufgaben des Fachs bzw. der Fachgruppe in der Schule vor dem Hintergrund der Schülerschaft**

Beim Gymnasium Dionysianum Rheine handelt es sich um eine drei- bis vierzügige Schule in Rheine mit zurzeit ca. 700 Schülerinnen und Schülern, 42 Planstellen und 59 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst weite Teile Rheines sowie umliegender Städte, was zum Teil auf das Angebot der Schule im Fach Informatik zurückzuführen ist. Im Bereich der Sekundarstufe II kooperiert das Gymnasium Dionysianum mit dem Emsland-Gymnasium und bietet mit ihm zahlreiche gemeinsame Kurse an, unter anderem in Informatik. Daher sind im Fach Informatik enge Absprachen mit der Kooperationsschule erforderlich.

Das Fach Informatik wird am Gymnasium Dionysianum ab der Jahrgangsstufe EF dreistündig unterrichtet. Es baut nicht auf dem Wahlpflichtfach Informatik des Wahlpflichtbereichs II in den Stufe 8 und 9 oder auf dem Profilangebot NT (Neue Technologie) der informatischen Grundbildung der Stufe 5 bis 7 auf.

## **1.3. Funktionen und Aufgaben der Fachgruppe vor dem Hintergrund des Schulprogramms**

In der Sekundarstufe II bietet das Gymnasium Dionysianum das Fach Informatik in Grund- und Leistungskursen an. Diese werden in der Regel beide eingerichtet. Es existieren daher längere Erfahrungen mit dem Unterricht und den Abiturprüfungen in Grund- und Leistungskursen.

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 prozeduralen Programmiersprache Java und der deklarativen Datenbanksprache SQL durchgeführt. Auch Ausflüge in die Assembler-Programmierung werden im Bereich eines modellierten von-Neumann-Rechners (MOPS) gemacht. In der Einführungsphase kommt dabei zusätzlich eine *didaktische Bibliothek* zum Einsatz, welche das Erstellen von grafischen Programmen erleichtert. Der Einsatz eines Schulbuches (Informatik, Lehrwerk für die gymnasiale Oberstufe. Schönigh) in allen Jahrgangsstufen der Sekundarstufe II erleichtert den SuS die häusliche Vor- und Nachbereitung des Unterrichts und sorgt für eine breite Basis an Informationstexten, Beispielen und Aufgabenstellungen.

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, da eine Binnendifferenzierung schon der notwendigen Unterrichtsgestaltung im Fach Informatik inhärent ist. Hier wird auch die **Vernetzung** mit anderen Fächern (z.B. Physik bzgl. Quantencomputern) oder Philosophie (z.B. Verhalten und Bewertung von KI-Algorithmen, autonomen Werkzeugen wie „Killer“-drohnen) und Sozialwissenschaften / STUBO bzgl. der **Digitalisierung der Arbeits- und Lebenswelt** (Welche Berufsbilder werden verschwinden?) gesucht.

Der Austausch von Unterrichtsmaterial und Klausuren, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik in Absprache mit dem Kooperationsgymnasium Emsland-Gymnasium stellen einen wichtigen Beitrag zur Qualitätssicherung und Entwicklung des Unterrichts dar.

Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor. Leistungskurse und Grundkurse in der Kooperation werden ausschließlich in Doppelstunden unterrichtet, wodurch ein regelmäßiger Wechsel zwischen vier und sechs (in Leistungskursen) oder zwei und vier (in Grundkursen) Wochenstunden erfolgt.

Der Informatikunterricht fördert den Bildungsprozess der Schülerinnen und Schüler in seiner personalen, sozialen und fachlichen Dimension. Bildung wird dabei als Lern- und Entwicklungsprozess verstanden, der sich auf das Individuum bezieht und in dem kognitives und emotionales, fachliches und fachübergreifendes Lernen, individuelle und soziale Erfahrungen, Theorie und Praxis miteinander verknüpft und auch ethische Kategorien (Informatiksysteme, Informatik, Mensch und Gesellschaft: Umgang mit den eigenen Daten) vermittelt und angeeignet werden. Gerade auch der Informatikunterricht am Gymnasium soll die sprachliche Kommunikationsfähigkeit des heranwachsenden jungen Menschen entwickeln und erweitern helfen, damit der einzelne sich in einer demokratischen, sich ständig verändernden Gesellschaft als autonomes und zur Mitbestimmung fähiges Individuum behaupten und bewähren kann.

Der Informatikunterricht schließt hier an die Präambel des Schulprogramms an:

„Dieses Ziel wollen wir durch **guten Unterricht** erreichen. Wir vermitteln

- *fachbezogenes Basiswissen*
- *fächerübergreifendes Orientierungswissen*
- *methodische Fertigkeiten in allen Lernbereichen*
- *Teamfähigkeit und Kooperationsbereitschaft*
- *als zentrale Werte Rücksicht, Toleranz und Fairness*

Dies zeigt sich, wenn **Wertschätzung** aller am Unterricht Beteiligten da ist, wenn **Soziale und personale Kompetenz** aufgebaut und **Medien-, Methoden- und Fachkompetenz** entwickelt und gesichert werden. Unterricht ist hierbei alles, was Kompetenzen stärkt!“

Verstehens -, Darstellungs- und Reflexionskompetenz werden auf der Grundlage der in der Sek. I vermittelten Kenntnisse und Einsichten in kontinuierlicher Progression in der Oberstufe weiterentwickelt.

Im Informatikunterricht der Sek. II erwerben die Schülerinnen und Schüler Fähigkeiten in den Bereichen Modellieren, Implementieren, Darstellen und Interpretieren, Argumentieren. Ergänzt werden die genannten Bereiche durch den Bereich Methodenkompetenz im Sinne von Kommunizieren und Kooperieren. Diese Bereiche strukturieren auch die fachliche Arbeit in der Oberstufe und genügen dem **Medienkompetenzrahmen NRW**.

1. BEDIENEN UND ANWENDEN	2. INFORMIEREN UND RECHERCHIEREN	3. KOMMUNIZIEREN UND KOOPERIEREN	4. PRODUZIEREN UND PRÄSENTIEREN	5. ANALYSIEREN UND REFLEKTIEREN	6. PROBLEMLÖSEN UND MODELLIEREN
<b>1.1 Medienausstattung (Hardware)</b> Medienausstattung (Hardware) kennen, auswählen und reflektiert anwenden; mit dieser verantwortungsvoll umgehen	<b>2.1 Informationsrecherche</b> Informationsrecherchen zielgerichtet durchführen und dabei Suchstrategien anwenden	<b>3.1 Kommunikations- und Kooperationsprozesse</b> Kommunikations- und Kooperationsprozesse mit digitalen Werkzeugen zielgerichtet gestalten sowie mediale Produkte und Informationen teilen	<b>4.1 Medienproduktion und Präsentation</b> Medienprodukte adressatengerecht planen, gestalten und präsentieren; Möglichkeiten des Veröffentlichens und Teilens kennen und nutzen	<b>5.1 Medienanalyse</b> Die Vielfalt der Medien, ihre Entwicklung und Bedeutungen kennen, analysieren und reflektieren	<b>6.1 Prinzipien der digitalen Welt</b> Grundlegende Prinzipien und Funktionsweisen der digitalen Welt identifizieren, kennen, verstehen und bewusst nutzen
<b>1.2 Digitale Werkzeuge</b> Verschiedene digitale Werkzeuge und deren Funktionsumfang kennen, auswählen sowie diese kreativ, reflektiert und zielgerichtet einsetzen	<b>2.2 Informationsauswertung</b> Themenrelevante Informationen und Daten aus Medienangeboten filtern, strukturieren, umwandeln und aufbereiten	<b>3.2 Kommunikations- und Kooperationsregeln</b> Regeln für digitale Kommunikation und Kooperation kennen, formulieren und einhalten	<b>4.2 Gestaltungsmittel</b> Gestaltungsmittel von Medienprodukten kennen, reflektiert anwenden sowie hinsichtlich ihrer Qualität, Wirkung und Aussageabsicht beurteilen	<b>5.2 Meinungsbildung</b> Die interessengeleitete Setzung und Verbreitung von Themen in Medien erkennen sowie in Bezug auf die Meinungsbildung beurteilen	<b>6.2 Algorithmen erkennen</b> Algorithmische Muster und Strukturen in verschiedenen Kontexten erkennen, nachvollziehen und reflektieren
<b>1.3 Datenorganisation</b> Informationen und Daten sicher speichern, wiederfinden und von verschiedenen Orten abrufen; Informationen und Daten zusammenfassen, organisieren und strukturiert aufbewahren	<b>2.3 Informationsbewertung</b> Informationen, Daten und ihre Quellen sowie dahinterliegende Strategien und Absichten erkennen und kritisch bewerten	<b>3.3 Kommunikation und Kooperation in der Gesellschaft</b> Kommunikations- und Kooperationsprozesse im Sinne einer aktiven Teilhabe an der Gesellschaft gestalten und reflektieren; ethische Grundsätze sowie kulturell-gesellschaftliche Normen beachten	<b>4.3 Quellendokumentation</b> Standards der Quellenangaben beim Produzieren und Präsentieren von eigenen und fremden Inhalten kennen und anwenden	<b>5.3 Identitätsbildung</b> Chancen und Herausforderungen von Medien für die Realitätswahrnehmung erkennen und analysieren sowie für die eigene Identitätsbildung nutzen	<b>6.3 Modellieren und Programmieren</b> Probleme formalisiert beschreiben, Problemlösestrategien entwickeln und dazu eine strukturierte, algorithmische Sequenz planen; diese auch durch Programmieren umsetzen und die gefundene Lösungsstrategie beurteilen
<b>1.4 Datenschutz und Informationssicherheit</b> Verantwortungsvoll mit persönlichen und fremden Daten umgehen; Datenschutz, Privatsphäre und Informationssicherheit beachten	<b>2.4 Informationskritik</b> Unangemessene und gefährdende Medieninhalte erkennen und hinsichtlich rechtlicher Grundlagen sowie gesellschaftlicher Normen und Werte einschätzen; Jugend- und Verbraucherschutz kennen und Hilfs- und Unterstützungsstrukturen nutzen	<b>3.4 Cybergewalt und -kriminalität</b> Persönliche, gesellschaftliche und wirtschaftliche Risiken und Auswirkungen von Cybergewalt und -kriminalität erkennen sowie Ansprechpartner und Reaktionsmöglichkeiten kennen und nutzen	<b>4.4 Rechtliche Grundlagen</b> Rechtliche Grundlagen des Persönlichkeits- (u.a. des Bildrechts), Urheber- und Nutzungsrechts (u.a. Lizenzen) überprüfen, bewerten und beachten	<b>5.4 Selbstregulierte Mediennutzung</b> Medien und ihre Wirkungen beschreiben, kritisch reflektieren und deren Nutzung selbstverantwortlich regulieren; andere bei ihrer Mediennutzung unterstützen	<b>6.4 Bedeutung von Algorithmen</b> Einflüsse von Algorithmen und Auswirkung der Automatisierung von Prozessen in der digitalen Welt beschreiben und reflektieren

Im Rahmen der **Verbraucherbildung** wird gerade auf die Gefahren beim Umgang mit schützenswerten Personendaten mit Blick auf die DSGVO eingegangen: Beispielhaft sind hier die Unterrichtsreihen zu Datenbanken und Netzwerken (Kryptologie) bzgl. Verschlüsselung und Kennworten zu nennen.

## 1.4. Verfügbare Ressourcen

Die Fachkonferenz besteht aus 0 Lehrerinnen und 3 Lehrern. Alle Kollegen haben die SI und SII Fakultas. Das zur Verfügung stehende Stundenvolumen beschränkt sie auf ca. 22 Std., da Herr Meer als Schulleiter nur mit einem geringen Stundenanteil zur Verfügung steht. Jedoch ist die Erteilung von Leistungskursen oder auch die Wahl von Informatik als drittes Abitur uneingeschränkt möglich, da bei drei Fachkollegen bei längerer Erkrankung schulinterner Ersatz möglich ist.

Es gibt zwei Fachräume mit je 32 vernetzten Computerarbeitsplätzen angebunden an die Serverstruktur des Pädagogischen Netzes. Weiterhin besitzt die Informatik ein eigenes WLAN für Handypogrammierung z.B. über den MIT-APP-Inventor

## 1.5. Funktionsinhaber/innen der Fachgruppe

Die Fachkonferenz wird von Herrn Dr. Janssen (VZ) und Herrn Meer (Stellver. VZ) geleitet.

---

## 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, des Medienkompetenzrahmenplans NRW und der Verbraucherbildung von ihnen erfüllt werden können.

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

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.2) 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.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.3) Beispiele und Materialien mit Bezug zum eingeführten Schulbuch, die 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, die im Einzelnen auch den Kapiteln 4.1 bis 4.2 zu entnehmen sind.

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

## 2.2. Übersichtsraster Unterrichtsvorhaben

### 2.2.1. Einführungsphase

Einführungsphase	
<p><b><u>Unterrichtsvorhaben E-I</u></b></p> <p><b>Thema:</b> <i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Einsatz, Nutzung und Aufbau von Informatiksystemen</li> <li>- Wirkung der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 6 Stunden</p>	<p><b><u>Unterrichtsvorhaben E-II</u></b></p> <p><b>Thema:</b> <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>
<p><b><u>Unterrichtsvorhaben E-III</u></b></p> <p><b>Thema:</b> <i>Algorithmische Grundstrukturen in Java</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Analyse, Entwurf und Implementierung einfacher Algorithmen</li> </ul> <p><b>Zeitbedarf:</b> 18 Stunden</p>	<p><b><u>Unterrichtsvorhaben E-IV</u></b></p> <p><b>Thema:</b> <i>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i></p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Kommunizieren und Kooperieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Informatiksysteme</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Binäre Codierung und Verarbeitung</li> <li>- Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten</li> </ul> <p><b>Zeitbedarf:</b> 8 Stunden</p>

## Einführungsphase

### Unterrichtsvorhaben E-V

**Thema:**

*Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele*

**Zentrale Kompetenzen:**

- Kommunizieren und Kooperieren
- Darstellen und Interpretieren
- Argumentieren
- Modellieren
- Implementieren

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Syntax und Semantik einer Programmiersprache
- Analyse, Entwurf und Implementierung einfacher Algorithmen

**Zeitbedarf:** 18 Stunden

### Unterrichtsvorhaben E-VI

**Thema:**

*Such- und Sortieralgorithmen*

**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Algorithmen
- Daten und ihre Strukturierung

**Inhaltliche Schwerpunkte:**

- Algorithmen zum Suchen und Sortieren
- Analyse, Entwurf und Implementierung einfacher Algorithmen
- Objekte und Klassen

**Zeitbedarf:** 9 Stunden

### Unterrichtsvorhaben E-VII

**Thema:**

*Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?*

**Zentrale Kompetenzen:**

- Kommunizieren und Kooperieren
- Darstellen und Interpretieren
- Argumentieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Geschichte der automatischen Datenverarbeitung
- Wirkungen der Automatisierung
- Dateisystem

**Zeitbedarf:** 10 Stunden

**Summe Einführungsphase: 77 Stunden**

## 2.2.2. Qualifikationsphase (Q1 und Q2)

In dieser Übersicht sind die zusätzlichen Inhalte für den Leistungskurs besonders gekennzeichnet. Weitere Vertiefungen im Leistungskurs ergeben sich durch den Anspruch der Probleme, den Umfang der Unterrichtsvorhaben sowie den Grad der Selbstständigkeit der SuS.

Qualifikationsphase – Q1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p><b>Thema:</b> Wiederholung und Vertiefung der objekt-orientierten Modellierung</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Darstellen und Interpretieren</li> <li>- Implementieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> <li>- Informatik, Mensch und Gesellschaft</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Wirkung der Automatisierung</li> </ul> <p><b>Zeitbedarf:</b> 14 Stunden (GK); 24 Stunden (LK)</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p><b>Thema:</b> Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Modellieren</li> <li>- Implementieren</li> <li>- Darstellen und Interpretieren</li> <li>- Argumentieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Daten und ihre Strukturierung</li> <li>- Algorithmen</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Objekte und Klassen</li> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden (GK); 40 Stunden (LK)</p>
<p><u>Unterrichtsvorhaben Q1-III</u></p> <p><b>Thema:</b> Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Implementieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Algorithmen</li> <li>- Formale Sprachen und Automaten</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Analyse, Entwurf und Implementierung von Algorithmen</li> <li>- Algorithmen in ausgewählten informatischen Kontexten</li> <li>- Syntax und Semantik einer Programmiersprache</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden (GK); 30 Stunden (LK)</p>	<p><u>Unterrichtsvorhaben Q1-IV</u></p> <p><b>Thema:</b> Automaten und formale Sprachen</p> <p><b>Zentrale Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>- Argumentieren</li> <li>- Darstellen und Interpretieren</li> <li>- Modellieren</li> <li>- Kommunizieren und Kooperieren</li> </ul> <p><b>Inhaltsfelder:</b></p> <ul style="list-style-type: none"> <li>- Formale Sprachen und Automaten</li> <li>- Informatiksysteme</li> </ul> <p><b>Inhaltliche Schwerpunkte:</b></p> <ul style="list-style-type: none"> <li>- Syntax und Semantik einer Programmiersprache</li> <li>- Endliche Automaten</li> <li>- Grammatiken regulärer Sprachen</li> <li>- Möglichkeiten und Grenzen von Automaten und formalen Sprachen</li> <li>- LK: Kellerautomaten</li> <li>- LK: Grammatiken kontextfreier Sprachen</li> <li>- LK: Scanner, Parser, Interpreter für reguläre Sprache</li> </ul> <p><b>Zeitbedarf:</b> 20 Stunden (GK); 30 Stunden (LK)</p>
<p>Summe Qualifikationsphase 1: 74 Stunden (GK); 124 Stunden (LK)</p>	

## Qualifikationsphase – Q2

### Unterrichtsvorhaben Q2-I

**Thema:** Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Modellieren
- Implementieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

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

**Inhaltliche Schwerpunkte:**

- Objekte und Klassen
- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache

**Zeitbedarf:** 20 Stunden (GK); 36 Stunden (LK)

### Unterrichtsvorhaben Q2-II

**Thema:** Aufbau von und Kommunikation in Netzwerken

**Zentrale Kompetenzen:**

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen, Wirkungen der Automatisierung

**Zeitbedarf:** 16 Stunden (GK); 24 Stunden (LK)

### Unterrichtsvorhaben Q2-III

**Thema:** Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

**Zentrale Kompetenzen:**

- Argumentieren
- Modellieren
- Implementieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

**Inhaltsfelder:**

- Daten und ihre Strukturierung
- Algorithmen
- Formale Sprachen und Automaten
- Informatik, Mensch und Gesellschaft

**Inhaltliche Schwerpunkte:**

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Sicherheit

**Zeitbedarf:** 20 Stunden (GK); 30 Stunden (LK)

### Unterrichtsvorhaben Q2-IV

**Thema:** Wiederholung ausgewählter Aspekte

**Zentrale Kompetenzen:**

- nach Bedarf der SuS

**Inhaltsfelder:**

- nach Bedarf der SuS

**Inhaltliche Schwerpunkte:**

- nach Bedarf der SuS

**Zeitbedarf:** nach Bedarf der SuS

**Summe Qualifikationsphase 2: 56 Stunden (GK); 90 Stunden (LK)**

---

## 2.3. Konkretisierte Unterrichtsvorhaben

Im Folgenden sollen die im *Unterkapitel 2.2.* aufgeführten Unterrichtsvorhaben konkretisiert werden.

Hinweis:

Verbindliche Festlegungen der Fachkonferenz:

Die Fachkonferenz des Gymnasium Dionysianum hat Themen, Leitfragen und die Ausführungen unter der Überschrift Vorhabenbezogene Konkretisierung verbindlich vereinbart, ebenso die Sequenzierung der Unterrichtsvorhaben (erste Tabellenspalte) und die ausgewiesenen Kompetenzen (zweite Tabellenspalte). Alle Mitglieder der Fachkonferenz haben sich darauf verständigt, in ihrem Unterricht Lerngelegenheiten anzubieten, so dass Schülerinnen und Schüler diese Kompetenzen im Rahmen der festgelegten Unterrichtssequenzen erwerben oder vertiefen können.

Unterrichtliche Anregungen:

Die angeführten Beispiele, Medien und Materialien sind dagegen Vorschläge bzw. Hilfen für die Lehrkräfte des Gymnasium Dionysianum mit Bezug auf das eingeführte Lehrwerk „Informatik 1“ und „Informatik 2“ aus dem Schöningh-Verlag. In diesen Bereichen sind Abweichungen von den vorgeschlagenen Vorgehensweisen und Ergänzungen möglich.

Zur Einführung in die objektorientierte Programmierung wird die didaktische Lernumgebung Greenfoot gewählt. Die Lehrtexte und Aufgaben des Lehrwerks „Informatik 1“ beziehen sich auch speziell auf ein bestimmtes Greenfoot-Szenario (Roboter).

Nach der Einarbeitungsphase sollte auf eine komplexere Lernumgebung umgestellt werden. Hier bieten sich verschiedene Möglichkeiten (in alphabetischer Reihenfolge) an: BlueJ, Eclipse, Java Editor oder NetBeans.

Im Lehrwerk sind die Lehrtexte und Aufgaben so verfasst, dass jede der Umgebungen eingesetzt werden kann. Eine Auswahl der Lernumgebung erfolgt auch unter Berücksichtigung der Lauffähigkeit und des Administrationsaufwands im pädagogischen Netzwerk.

In der Qualifikationsphase werden die Unterrichtsvorhaben unter Berücksichtigung der Vorgaben für das Zentralabitur Informatik in NRW konkretisiert. Diese sind zu beziehen unter der Adresse

<https://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/index.html> (abgerufen: 05.12.2019)

### 2.3.1. Einführungsphase

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

Die Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),
- präsentieren Arbeitsabläufe und -ergebnisse (K),
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K).

### 2.3.1.1. Unterrichtsvorhaben EF-I

**Thema:** Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik

**Leitfragen:** Was macht Informatik? Welche fundamentalen Konzepte müssen Informatikerinnen und Informatiker in ihre Arbeit einbeziehen, damit informatische Systeme effizient und zuverlässig arbeiten können? Wo lassen sich diese Konzepte (in Ansätzen) in dem schuleigenen Netzwerk- und Computersystem wiederfinden?

**Vorhabenbezogene Konkretisierung:**

Im ersten Unterrichtsvorhaben werden die fünf Inhaltsfelder des Faches Informatik beispielhaft an einem Informatiksystem erarbeitet. Das Unterrichtsvorhaben ist so strukturiert, dass die Schülerinnen und Schüler anhand bekannter Alltagstechnik die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) größtenteils selbstständig erarbeiten und nachvollziehen.

Ausgehend von dem bekannten Bedienungs- und Funktionalitätswissen eines Navigationsgerätes werden die Strukturierung von Daten, das Prinzip der Algorithmik, die Eigenheit formaler Sprachen, die Kommunikationsfähigkeit von Informatiksystemen und die positiven und negativen Auswirkungen auf Mensch und Gesellschaft thematisiert. Das am Navigationsgerät erworbene Wissen kann auf weitere den Schülerinnen und Schülern bekannte Informatiksysteme übertragen werden.

In einem letzten Schritt kann ausgehend von den Inhaltsfeldern das Schulnetzwerk in Ansätzen so analysiert werden, dass ein kompetenter Umgang mit diesem ermöglicht wird.

**Zeitbedarf:** 6 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Informatiksysteme und ihr genereller Aufbau</b> (a) Daten und ihre Strukturierung (b) Algorithmen (c) Formale Sprachen und Automaten (d) Informatiksysteme (e) Informatik, Mensch und Gesellschaft	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D)	<b>Kapitel 1 „Was macht Informatik“</b> Als Anschauungsmaterial bieten sich Navigationsgeräte an  MKR 2.x und MKR 5.x
<b>2. Der kompetente Umgang mit dem Schulnetzwerk</b> (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordnern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		<b>Kapitel 1 „Was macht Informatik“</b> Interview mit dem Netzwerkadministrator, Benutzer- und Datenschutzbestimmungen der Schule  MKR1.1 bis 1.4

## 2.3.1.2. Unterrichtsvorhaben EF-II

**Thema:** Grundlagen der objektorientierten Analyse, Modellierung und Implementierung

**Leitfragen:** *Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren?*

### **Vorhabenbezogene Konkretisierung:**

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung in diesem Kontext ein.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung und Modellierungswerkzeuge wie Objektdiagramme und Klassendiagramme eingeführt.

Im Anschluss wird die objektorientierte Analyse für das Greenfoot-Szenario Planetenerkundung durchgeführt. Die vom Szenario vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt (Kapitel 4) zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden.

Da in Kapitel 2 zudem auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben (Kapitel 3) führen. Dort stehen Kontrollstrukturen im Mittelpunkt.

**Zeitbedarf:** 8 Stunden

### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Identifikation von Objekten und Klassen</b> (a) An einem lebensweltnahen Beispiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (b) Objekte werden durch Objektdiagramme, Klassen durch Klassendiagramme dargestellt. (c) Die Modellierungen werden einem konkreten Anwendungsfall entsprechend angepasst.	Die Schülerinnen und Schüler - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften und ihre Operationen (M), - stellen den Zustand eines Objekts dar (D), - modellieren Klassen mit ihren Attributen und ihren Methoden (M), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - implementieren Klassen in einer Program-	<b>Kapitel 2 Einführung in die Objektorientierung</b> 2.1 Objektorientierte Modellierung  MKR 6.3

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>2. Analyse von Objekten und Klassen im Greenfoot-Szenario</b>            (a) Schritte der objektorientierten Analyse, Modellierung und Implementation            (b) Analyse und Erprobung der Objekte im Greenfoot-Szenario</p>	<p>miersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).</p>	<p><b>Kapitel 2 Einführung in die Objektorientierung</b>            2.2 Das Greenfoot-Szenario „Planetenerkundung“            Von der Realität zu Objekten            Von den Objekten zu Klassen, Klassendokumentation            Objekte inspizieren            Methoden aufrufen            Objektidentität und Objektzustand</p> <p>MKR 6.2. und MKR 6.3</p>
<p><b>3. Implementierung einfacher Aktionen in Greenfoot</b>            (a) Quelltext einer Java-Klasse            (b) Implementation eigener Methoden, Dokumentation mit JavaDoc            (c) Programme übersetzen (Aufgabe des Compilers) und testen</p>		<p><b>Kapitel 2 Einführung in die Objektorientierung</b>            2.3 Programmierung in Greenfoot            Methoden schreiben            Programme übersetzen und testen</p> <p>MKR 6.3</p>

### 2.3.1.3. Unterrichtsvorhaben EF-III

**Thema:** Algorithmische Grundstrukturen in Java

**Leitfragen:** *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

#### **Vorhabenbezogene Konkretisierung:**

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt dabei auf der Erarbeitung von Kontrollstrukturen. Die Strukturen Wiederholung und bedingte Anweisung werden an einfachen Beispielen eingeführt und anschließend anhand komplexerer Problemstellungen erprobt. Da die zu entwickelnden Algorithmen zunehmend umfangreicher werden, werden systematische Vorgehensweisen zur Entwicklung von Algorithmen thematisiert.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

**Zeitbedarf:** 18 Stunden

#### **Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Algorithmen</b> (a) Wiederholungen (While-Schleife) (b) bedingte Anweisungen (c) Verknüpfung von Bedingungen durch die logischen Funktionen UND, ODER und NICHT (d) Systematisierung des Vorgehens zur Entwicklung von Algorithmen zur Lösung komplexerer Probleme	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 zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (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).	<b>Kapitel 3 Algorithmen</b> 3.1 Wiederholungen 3.2 Bedingte Anweisungen 3.3 Logische Funktionen 3.4 Algorithmen entwickeln  MKR 6.2 und MKR 6.3
<b>2. Variablen und Methoden</b> (a) Implementierung eigener Methoden mit lokalen Variablen, auch zur Realisierung einer Zählschleife (b) Implementierung eigener Methoden mit Parameterübergabe und/oder Rückgabewert (c) Implementierung von Konstruktoren (d) Realisierung von Attributen		<b>Kapitel 4 Variablen und Methoden</b> 4.1 lokale Variablen 4.2 Methoden 4.3 Attribute  MKR 6.2 und MKR 6.3

### 2.3.1.4. Unterrichtsvorhaben EF-IV

**Thema:** Das ist die digitale Welt! - Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung

**Leitfragen:** Wie werden binäre Informationen gespeichert und wie können sie davon ausgehend weiter verarbeitet werden? Wie unterscheiden sich analoge Medien und Geräte von digitalen Medien und Geräten? Wie ist der Grundaufbau einer digitalen Rechenmaschine?

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben hat die binäre Speicherung und Verarbeitung sowie deren Besonderheiten zum Inhalt.

Im ersten Schritt erarbeiten die Schülerinnen und Schüler anhand ihnen bekannter technischer Gegenstände die Gemeinsamkeiten, Unterschiede und Besonderheiten der jeweiligen analogen und digitalen Version. Nach dieser ersten grundlegenden Einordnung des digitalen Prinzips wenden die Schülerinnen und Schüler das Binäre als Zahlensystem mit arithmetischen und logischen Operationen an und codieren Zeichen binär.

Zum Abschluss soll der grundlegende Aufbau eines Rechnersystems im Sinne der von-Neumann-Architektur erarbeitet werden.

**Zeitbedarf:** 8 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<b>1. Analoge und digitale Aufbereitung und Verarbeitung von Daten</b> (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die Schülerinnen und Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) - stellen ganze Zahlen und Zeichen in Binärcodes dar (D), - interpretieren Binärcodes als Zahlen und Zeichen (D)	<b>Exkurs „Analog und Digital“</b>  MKR 2.x, MKR 3.3 u. 3.4
<b>2. Der Umgang mit binärer Codierung von Informationen</b> (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärrechners	- beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	<b>Exkurs „Binäre Welt“</b>  MKR 6.1
<b>3. Aufbau informatischer Systeme</b> (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips		<b>Exkurs „Arbeitsweise eines Computers“</b>  MKR 6.1

### 2.3.1.5. Unterrichtsvorhaben EF-V

**Thema:** Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele

**Leitfragen:** Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben hat die Entwicklung von Objekt -und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Ausgehend von der Dekonstruktion und Erweiterung eines Spiels wird ein weiteres Projekt von Grund auf modelliert und implementiert. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

**Zeitbedarf:** 18 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Umsetzung von Anforderungen in Entwurfsdiagramme</b> (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	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),	<b>Kapitel 5 Klassenentwurf</b> 5.1. Von der Realität zum Programm 5.2. Objekte identifizieren 5.3. Klassen und Beziehungen entwerfen  MKR 6.3

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>2. Implementationsdiagramme als erster Schritt der Programmierung</b></p> <p>(a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden</p> <p>(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern</p> <p>(c) Entwicklung von Klassendokumentationen</p> <p>(d) Erstellung von Sequenzdiagrammen als Vorbereitung Vorbereitung für die Programmierung</p>	<ul style="list-style-type: none"> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M),</li> <li>- ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),</li> <li>- modellieren Klassen unter Verwendung von Vererbung (M),</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I),</li> <li>- testen Programme schrittweise anhand von Beispielen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> </ul>	<p><b>Kapitel 5 Klassenentwurf</b></p> <p>5.4 Klassen und Beziehungen implementieren</p> <p>5.5 Vererbung</p> <p>MKR 6.3</p>
<p><b>3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes</b></p> <p>(a) Klassen werden in Java-Quellcode umgesetzt</p> <p>(b) Das Geheimnisprinzip wird umgesetzt</p> <p>(c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.</p>	<ul style="list-style-type: none"> <li>- analysieren und erläutern einfache Algorithmen und Programme (A)</li> <li>- modifizieren einfache Algorithmen und Programme (I),</li> <li>- entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M).</li> <li>- stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),</li> <li>- dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)</li> </ul>	<p><b>Kapitel 5 Klassenentwurf</b></p> <p>5.4 Klassen und Beziehungen implementieren</p> <p>5.5 Vererbung</p> <p>MKR 6.3</p>
<p><b>4. Vererbungsbeziehungen</b></p> <p>(a) Das Grundprinzip der Vererbung wird erarbeitet</p> <p>(b) Die Vorteile der Vererbungsbeziehungen</p> <p>(c) Vererbung wird implementiert</p>		<p><b>Kapitel 5 Klassenentwurf</b></p> <p>5.5 Vererbung</p> <p>MKR 6.3</p>
<p><b>5. Softwareprojekt</b></p> <p>(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte)</p> <p>(b) Erweiterung des Spiels um weitere Funktionalitäten</p> <p>(c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche</p> <p>(d) (arbeitsteilige) Implementation des Spiels</p>		<p><b>Kapitel 8 Softwareprojekte</b></p> <p>8.1 Softwareentwicklung</p> <p>8.2 Oberflächen</p> <p>MKR 1.2, 1.3, 3.3, 6.1</p>

### 2.3.1.6. Unterrichtsvorhaben EF-VI

**Thema:** Such- und Sortieralgorithmen

**Leitfragen:** *Wie können Objekte bzw. Daten effizient gesucht und sortiert werden?*

**Vorhabenbezogene Konkretisierung:**

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortieralgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierverfahren beurteilt.

**Zeitbedarf:** 9 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

<b>Unterrichtssequenzen</b>	<b>Zu entwickelnde Kompetenzen</b>	<b>Kapitel und Materialien</b>
<b>1. Modellierung und Implementation von Datenansammlungen</b> (a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld	Die Schülerinnen und Schüler - analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) - entwerfen einen weiteren Algorithmus zum Sortieren (M) - beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A)	<b>Kapitel 7 Sortieren und Suchen auf Feldern</b> 7.1 Das Feld – Eine Sammlung von Daten  MKR 6.2
<b>2. Explorative Erarbeitung von Suchverfahren</b> (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und mithilfe einer Berechnungsfunktion. (b) Vergleich der drei Verfahren durch intuitive Effizienzbetrachtungen.	- ordnen Attributen lineare Datenansammlungen zu (M)	<b>Kapitel 7 Sortieren und Suchen auf Feldern</b> Projekteinstieg 1: Suchen 7.2 Suchen mit System Lineare Suche Binäre Suche Hashing  MKR 6.2

<p><b>3. Systematisierung von Algorithmen und Effizienzbetrachtungen</b></p> <p>(a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode</p> <p>(b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele</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>(e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)</p>		<p><b>Kapitel 7 Sortieren und Suchen auf Feldern</b></p> <p>Projekteinstieg 2: Sortieren</p> <p>7.3 Ordnung ist das halbe Leben!? – Sortieren</p> <p>Sortieren</p> <p>Selection Sort</p> <p>Insertion Sort</p> <p>Bubble Sort</p> <p>MKR 6.2 u. MKR 6.3</p>
--	--	--

### 2.3.1.7. Unterrichtsvorhaben EF-VII

**Thema:** Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

**Leitfragen:** *Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen?*

**Vorhabenbezogene Konkretisierung:**

Das Unterrichtsvorhaben stellt die verschiedenen Entwicklungsstränge der Informatik in den Fokus. Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Im ersten Schritt des Unterrichtsvorhabens wird anhand von Themenkomplexen entscheidende Entwicklungen der Informatik erarbeitet. Dabei werden auch übergeordnete Tendenzen identifiziert.

Ausgehend von dieser Betrachtung kann die aktuelle Informatik hinsichtlich ihrer Leistungsfähigkeit analysiert werden. Dabei soll herausgestellt werden, welche positiven und negativen Folgen Informatiksysteme mit sich bringen können.

**Zeitbedarf:** 12 Stunden

**Sequenzierung des Unterrichtsvorhabens:**

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Schriftzeichen, Rechenmaschine, Computer</b>            (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der Informatik vorgestellt werden.            (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),</li> <li>- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A)</li> <li>- nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)</li> </ul>	<p><i>Exkurs „Geschichte der Informatik“</i></p> <p>MKR 5.x</p>
<p><b>2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren</b>            (a) Ausgehend von 1. werden Tendenzen der Entwicklung der Informatik erarbeitet            (b) Informatik wird als Hilfswissenschaft klassifiziert, die weit über ihren originären Bereich hinaus Effizienz- und Leistungssteigerungen erzeugt            (c) Anhand von Fallbeispielen werden technische und organisatorische Vorteile, sowie deren datenschutzrechtlichen Nachteile betrachtet.</p>		<p><i>Exkurs „Informatik und Gesellschaft“</i></p> <p>MKR 2.x</p>

---

## 2.3.2. 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).

### 2.3.2.1. Unterrichtsvorhaben Q1-I

**Thema:** Wiederholung und Vertiefung der objektorientierten Modellierung

**Leitfragen:** Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

**Vorhabenbezogene Konkretisierung:**

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektividentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

**Zeitbedarf:** 14 Stunden (GK); 24 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung</b></p> <p>a) Sichtweise der objektorientierten Informatik auf die Welt</p> <p>b) OOP als informatikspezifische Modellierung der Realität</p> <p>c) Schritte der Softwareentwicklung</p>	<p>Die Schülerinnen und Schüler ...</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern objektorientierte Modellierungen (A),</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M),</li> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),</li> <li>- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> </ul>	<p>2.1 Die Welt ist voller Objekte Projekteinstieg: Klassentwurf – step by step</p> <p>MKR 6.3</p>
<p><b>2. Erweiterung der objektorientierten Programmierung</b></p> <p>a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm</p> <p>b) Objektkommunikation im Sequenzdiagramm</p> <p>c) Klassendokumentation</p> <p>d) Umsetzung von Teilen der Modellierung</p>	<ul style="list-style-type: none"> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I),</li> <li>- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</li> </ul>	<p>2.2 Gut geplant – Klassentwurf</p> <p>2.3 Vererbungshierarchien nutzen</p> <p>MKR 6.3</p>
<p><b>3. Mensch und Technik</b></p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<ul style="list-style-type: none"> <li>- dokumentieren Klassen (D),</li> <li>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</li> <li>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</li> </ul>	<p>Die digitale Welt 001 - Mensch und Technik</p> <p>MKR 6.1</p>
<p><b>4. Übung und Vertiefung der OOM / OOP</b></p>	<ul style="list-style-type: none"> <li>- 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).</li> </ul>	<p>Prüfungsvorbereitung</p>

---

### 2.3.2.2. Unterrichtsvorhaben Q1-2

**Thema:**

Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

**Leitfragen:**

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

**Vorhabenbezogene Konkretisierung:**

Ausgehend von einigen Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur *Queue* erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt (Polizeikontrolle), dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur *Stack*, die mithilfe eines einfachen Anwendungsszenarios eingeführt (Biber/Palindrom) wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der *Stack* als auch die *Queue* benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft (LED) wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

**Zeitbedarf:** 20 Stunden (GK); 40 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Die Datenstruktur Feld</b></p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zwei-dimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> <li>- erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A)</li> <li>- konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M)</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I)</li> <li>- testen und korrigieren Computerprogramme (I)</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D)</li> <li>- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D)</li> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D)</li> <li>- modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M)</li> <li>- ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M)</li> <li>- dokumentieren Klassen (D)</li> <li>- implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)</li> <li>- erläutern Operationen dynamischer (linearer und nichtlinearer) Datenstrukturen (A)</li> <li>- LK: implementieren Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (I)</li> </ul>	<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Anforderungen an eine Datenstruktur</li> <li>- Speichern mit Struktur</li> </ul> <p><i>neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)</i></p> <p>MKR 1.3</p>
<p><b>2. Die Datenstruktur Schlange</b></p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> <li>- Wer zuerst kommt</li> <li>- Objekte miteinander verketten</li> <li>- Verwaltung und Inhalt</li> <li>- Funktionen der Queue</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorge lassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist.</li> <li>- Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus dem Verkehr gewunken. Es werden so lange Fahrzeuge kontrolliert, bis eine gewisse Menge an Verstößen vorliegt oder Autos kontrolliert wurden.</li> </ul>	<p>MKR 6.2 u. MKR 6.3</p>

<p><b>3. Die Datenstruktur Stapel</b></p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i></p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsschreibung mit Objekten der Klasse <i>Queue</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>		<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Daten gut abgelegt – Stapel</li> <li>- Funktionen der Datenstruktur Stapel</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- Standardoperationen/Basiskompetenz (Stapel Münzen/CDs) zur Umsetzung der gegebenen Funktionen der Klasse <i>Stack</i></li> <li>- Biber und Teller: Es gibt große und kleine Biber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl <i>Queue</i> als auch <i>Stack</i> verwendet werden.</li> <li>- Palindrom: Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.</li> </ul> <p>MKR 6.2 u. MKR 6.3</p>
<p><b>4. Die Datenstruktur Liste</b></p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		<p><b>Kapitel:</b></p> <ul style="list-style-type: none"> <li>- Flexibel für alle Fälle – (die) lineare Liste</li> <li>- Funktionen der Datenstruktur Liste</li> </ul> <p><b>Aufgaben:</b></p> <ul style="list-style-type: none"> <li>- LEDs</li> <li>- Textzeilen verarbeiten</li> </ul> <p>MKR 6.2 u. MKR 6.3</p>
<p><b>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</b></p>		<p>Prüfungsvorbereitung</p>

*Projekteinstieg Heldenspiel*

*Mit dem Heldenspiel können alle im Kapitel behandelten Datenstrukturen erarbeitet werden. Das Spiel kann bis zu einem beliebigen Grad realisiert werden, sodass es sowohl als Einstieg als auch als ein umfassendes Projekt für lineare Datenstrukturen genutzt werden kann.*

MKR 3.1, 3.3 und 4.2

---

### 2.3.2.3. Unterrichtsvorhaben Q1-III

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

**Leitfragen:** Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?

**Vorhabenbezogene Konkretisierung:**

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmenentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt werden.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

**Zeitbedarf:** 20 Stunden (GK); 30 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Eigenschaften von Algorithmen</b></p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D),</li> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> </ul>	<p>4.1 Ohne Algorithmen läuft nichts</p> <p>4.2 Teile die Arbeit – rekursive Algorithmen</p> <p>MKR 6.2 u. MKR 6.3 / 6.4</p>
<p><b>2. Suchen in Listen und Arrays</b></p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I).</li> <li>- implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I),</li> <li>- beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),</li> </ul>	<p>4.3 Suchen – iterativ und rekursiv</p> <p>MKR 6.2 u. MKR 6.3</p>
<p><b>3. Sortieren auf Listen und Arrays</b></p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> </ul>	<p>4.4 Sortieren – iterativ und rekursiv</p> <p>MKR 6.2 u. MKR 6.3 / 6.4</p>

---

#### 2.3.2.4. Unterrichtsvorhaben QI-IV

**Thema:** Automaten und formale Sprachen

**Leitfragen:**

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen beschrieben werden und welche nicht?

**Vorhabenbezogene Konkretisierung:**

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt bei der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht.

**Zeitbedarf:** 20 Stunden (GK); 30 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Endliche Automaten</b></p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D)</li> <li>- analysieren und erläutern die Eigenschaften endlicher Automaten (LK: und Kellerautomaten) einschließlich ihres Verhaltens bei bestimmten Eingaben (A)</li> <li>- ermitteln die Sprache, die ein endlicher Automat (LK: und Kellerautomat) akzeptiert (D)</li> <li>- entwickeln und modifizieren zu einer Problemstellung endliche Automaten (LK: oder Kellerautomaten) (M)</li> <li>- stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D)</li> <li>- entwickeln zur Grammatik einer regulären (LK: oder kontextfreien) Sprache einen zugehörigen endlichen Automaten (LK: oder Kellerautomaten) (M)</li> </ul>	<p>Vom realen Automaten zum Modell  ProjektEinstieg: Schatzsuche  Der Mealy-Automat  Der erkennende endliche Automat  Wort und Sprache  MKR 6.3</p>
<p><b>2. Grammatiken regulärer Sprachen</b></p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> <li>- analysieren und erläutern Grammatiken regulärer (LK: und kontextfreier) Sprachen (A)</li> <li>- modifizieren Grammatiken regulärer (LK: und kontextfreier) Sprachen (M)</li> <li>- entwickeln zu einer regulären (LK: oder kontextfreien) Sprache eine Grammatik, die die Sprache erzeugt (M)</li> <li>- entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M)</li> </ul>	<p>Grammatiken regulärer Sprachen  MKR 6.2 u. MKR 6.3</p>
<p><b>3. LK: Kellerautomaten und Grammatiken kontextfreier Sprachen</b></p> <p>a) Erarbeitung der formalen Beschreibung eines Kellerautomaten und Abgrenzung zum endlichen Automaten</p> <p>b) Erarbeitung der formalen Beschreibung einer kontextfreien Grammatik und Abgrenzung von regulären Grammatiken</p>	<ul style="list-style-type: none"> <li>- beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D)</li> <li>- zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf (LK: und erläutern diese)</li> <li>- LK: modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I)</li> </ul>	<p>MKR 6.2 u. MKR 6.3</p>
<p><b>4. Übungen und Vertiefungen</b></p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>		<p>Prüfungsvorbereitung</p>
<p><i>ProjektEinstieg: Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben.</i></p> <p>MKR 6.1, MKR 2.4</p>		

---

### 2.3.2.5. Unterrichtsvorhaben Q2-I

**Thema:** Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

**Leitfragen:** Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

**Vorhabenbezogene Konkretisierung:**

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Der Anwendungskontext Ahnenbaum wird mithilfe der Klasse BinaryTree (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse BinaryTree thematisiert.

Möchte man Daten geordnet speichern, so bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces Item realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen BinarySearchTree und Item modelliert und (ggf. in Teilen) implementiert.

**Zeitbedarf:** 20 Stunden (GK); 36 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Aufbau von Baumstrukturen und Grundbegriffe</b></p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D),</li> <li>- erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A),</li> <li>- analysieren und erläutern Algorithmen und Programme (A),</li> <li>- stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).</li> </ul>	<p>6.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p> <p>MKR 6.2 u. MKR 6.3</p>
<p><b>2. Binäre Bäume</b></p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A),</li> <li>- ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),</li> <li>- ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),</li> <li>- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M),</li> <li>- verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M),</li> </ul>	<p>6.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p> <p>MKR 6.2 u. MKR 6.3</p>
<p><b>3. Binäre Suchbäume</b></p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> <li>- entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M),</li> <li>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I),</li> <li>- modifizieren Algorithmen und Programme (I),</li> <li>- nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I),</li> <li>- interpretieren Fehlermeldungen und korrigieren den Quellcode (I),</li> <li>- testen Programme systematisch anhand von Beispielen (I),</li> </ul>	<p>6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p> <p>MKR 6.2 u. MKR 6.3</p>

<p><b>3. LK: Graphen</b></p> <p>a) Operationen auf Graphen</p> <p>b) Modellierung eines Graphen in einem Anwendungskontext mit Hilfe der Klasse Graph</p> <p>c) Implementation einer Anwendung der Datenstruktur Graph</p> <p>d) Tiefensuche, Breitensuche, Backtracking</p> <p>e) Dijkstra-Algorithmus</p>		<p>MKR 6.2 u. MKR 6.3</p>	
<p><b>4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</b></p>		<p>Prüfungsvorbereitung</p>	

---

### 2.3.2.6. Unterrichtsvorhaben Q2-II

**Thema:** Aufbau von und Kommunikation in Netzwerken

**Leitfragen:** Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden?

**Vorhabenbezogene Konkretisierung:**

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

**Zeitbedarf:** 12 Stunden (GK); 24 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</b></p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A),</li> <li>- nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D),</li> <li>- analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A).</li> </ul>	<p>7.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p> <p>MKR 3.1</p>
<p><b>2. Aufbau von Netzwerken und Kommunikationsregeln</b></p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregeltete technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<ul style="list-style-type: none"> <li>- LK: analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (A),</li> <li>- LK: entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M).</li> </ul>	<p>7.2 Ohne Protokoll läuft nichts - Netzwerke</p> <p>MKR 1.3 u. 1.4 u. MKR 6.1</p>
<p><b>3. Aufgabenteilung in Netzwerken durch Server und Client</b></p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>(b) Protokolle zwischen Client und Server</p>		<p>7.3 Einer für alle – Client-Server-Struktur</p> <p>MKR 1.3 u. 1.4 u. MKR 6.1</p>
<p><b>4. Kryptologie</b></p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 100 – Kryptologie</p> <p>MKR 1.3 u. 1.4 u. MKR 6.1 MKR 3.4</p>
<p><b>5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken</b></p>		<p>Prüfungsvorbereitung</p>

---

### 2.3.2.7. Unterrichtsvorhaben Q2-III

**Thema:** Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

**Leitfragen:** Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

**Vorhabenbezogene Konkretisierung:**

Am Beispiel eines Online-Buchhandels werden der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schülerinnen und Schüler nehmen dabei zunächst die Sicht der Anwender an, die eine bestehende Datenbank beschreiben und analysieren und mithilfe von SQL-Abfragen Daten gezielt herausfiltern.

Mithilfe des Projekteinstiegs „Tabellen“ können bereits zu einem frühen Zeitpunkt des Unterrichtsvorhabens Redundanzen, Inkonsistenzen und Anomalien problematisiert werden.

Nachdem die Lernenden in der ersten Sequenz mit Datenbanken vertraut gemacht wurden, nehmen sie nun die Rolle der Entwickler an, indem sie selbst Datenbanken von Grund auf modellieren und das Modell in ein Relationenschema überführen. Sie arbeiten mit Entity-Relationship-Diagrammen, um Entitäten, zugehörige Attribute, Relationen und Kardinalitäten in Anwendungskontexten darzustellen. Gegebene ER-Diagramme werden analysiert, erläutert und modifiziert.

Der bereits in der ersten Sequenz problematisierte Begriff der Redundanz wird am Ende des Unterrichtsvorhabens wieder aufgegriffen, um die Normalisierung von Datenbanken zu thematisieren. Bestehende Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (so weit nötig) normalisiert.

**Zeitbedarf:** 20 Stunden (GK); 30 Stunden (LK)

**Sequenzierung des Unterrichtsvorhabens:** s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
<p><b>1. Nutzung von relationalen Datenbanken</b></p> <p>a) Aufbau von Datenbanksystemen und Grundbegriffe</p> <ul style="list-style-type: none"> <li>- Aufgaben und Eigenschaften eines Datenbanksystems</li> <li>- Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema</li> <li>- Problematisierung von Redundanzen, Anomalien und Inkonsistenzen</li> </ul> <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> <li>- Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN)</li> <li>- Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, &lt;&gt;, &gt;, &lt;, &gt;=, &lt;=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke)</li> </ul> <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> <li>- erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A),</li> <li>- analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A),</li> <li>- analysieren und erläutern eine Datenbankmodellierung (A),</li> <li>- erläutern die Eigenschaften normalisierter Datenbankschemata (A),</li> <li>- bestimmen Primär- und Sekundärschlüssel (M),</li> <li>- ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M),</li> <li>- modifizieren eine Datenbankmodellierung (M),</li> <li>- modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M),</li> <li>- überführen Datenbankschemata in vorgegebene Normalformen (M),</li> <li>- verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I),</li> <li>- ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D),</li> <li>- stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D),</li> <li>- überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D),</li> <li>- LK: implementieren ein relationales Datenbankschema als Datenbank (I),</li> </ul>	<p><b>Kapitel 8 Datenbanken</b></p> <p>8.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>8.2 Daten anordnen mit Tabellen</p> <p>Beispiel: Buchhandlung</p> <p>Redundanzen, Anomalien und Inkonsistenzen</p> <p>Projekteinstieg: Tabellen</p> <p>8.3 Daten filtern mit SQL</p> <p>8.4 Komplexe Filter</p> <p>Aufgaben</p> <p>MKR 1.2, 1.3 u. 1.4 u. MKR 6.1</p>

<p><b>2. Modellierung von relationalen Datenbanken</b></p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> <li>- Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms</li> <li>- Erläuterung und Erweiterung einer Datenbankmodellierung</li> </ul> <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> <li>- Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln</li> </ul> <p>c) Normalformen</p> <ul style="list-style-type: none"> <li>- Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)</li> </ul> <p>d) Datenschutz und Datensicherheit mit Blick auf die DSGVO und Cyberkriminalität</p>		<p>8.5 Datenbankentwurf</p> <p><i>Beispiel: Online-Buchhandel</i></p> <p><i>Datenanalyse und Entwurf</i></p> <p>8.6 Umsetzung des ER-Modells</p> <p><i>Entitätsmengen</i></p> <p><i>m:n-Beziehungen</i></p> <p><i>1:n-Beziehungen</i></p> <p><i>1:1-Beziehungen</i></p> <p><i>Wiederaufgriff des Projekteinstiegs</i></p> <p><i>8.7 Datenbanken verbessern durch Normalformen</i></p> <p>MKR 4.4</p> <p>MKR 3.3 u. 3.4</p>	
<p><b>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</b></p>		<p>Prüfungsvorbereitung</p>	

---

### 2.3.2.8. Unterrichtsvorhaben Q2-IV

**Thema:** Wiederholung ausgewählter Aspekte

**Leitfragen:** Welche Inhaltsfelder aus der gymnasialen Oberstufe sollten noch vertieft werden? Welche Kompetenzen sollten die SuS noch weiter ausbauen?

**Vorhabenbezogene Konkretisierung:**

Ein Teil der am Ende der Q2 zur Verfügung stehenden Zeit wird für die Wiederholung von Teilaspekten ausgewählter Unterrichtsvorhaben verwendet. Dabei werden die Wünsche der SuS berücksichtigt. Ziel ist die vertiefende Vorbereitung der SuS auf die Abiturprüfung.

Nicht an der Abiturprüfung teilnehmende SuS können neben der Wiederholung auch an zusätzlichen über diesen Lehrplan hinausgehenden Inhalten arbeiten.

**Zeitbedarf:** nach Bedarf der SuS

**Sequenzierung des Unterrichtsvorhabens:** nach Bedarf der SuS

---

### 3. Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik des Gymnasium Dionysianum 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 22 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.
- 22) Informatik ist ein Methodenfach. Diesem Gedanken wird durch die Methodenvielfalt und die Reflexion der verwendeten Fachmethoden Rechnung getragen.

---

## 3.1. Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz des Gymnasium Dionysianum im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

### 3.1.1. Beurteilungsbereich Klausuren

#### Verbindliche Absprachen:

Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

#### Instrumente:

- Einführungsphase: 1 Klausur je Halbjahr  
Dauer der Klausur: 2 Unterrichtsstunden
- Grundkurse Q 1: 2 Klausuren je Halbjahr  
Dauer der Klausuren: 2 Unterrichtsstunden
- Grundkurse Q 2.1: 2 Klausuren  
Dauer der Klausuren: 3 Unterrichtsstunden
- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen

Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q1.2 eine **Facharbeit** geschrieben werden.

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

#### Kriterien

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

Spätestens ab der Qualifikationsphase orientiert sich die Zuordnung der Hilfspunktsumme zu den Notenstufen an dem Zuordnungsschema des Zentralabiturs.

Von diesem kann aber im Einzelfall begründet abgewichen werden, wenn sich z.B. besonders originelle Teillösungen nicht durch Hilfspunkte gemäß den Kriterien des Erwartungshorizontes abbilden lassen oder eine Abwertung wegen besonders schwacher Darstellung (APO-GOST §13 (2)) angemessen erscheint.

Die Note ausreichend (5 Punkte) soll bei Erreichen von 45 % der Hilfspunkte erteilt werden.

### 3.1.2. Beurteilungsbereich Sonstige Mitarbeit

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn des Schuljahres genannt.

---

### 3.1.2.1. Verbindliche Absprachen der Fachkonferenz

Falls nach der Behandlung der verbindlich genannten Inhalte noch ausreichend Zeit zur Verfügung steht, soll die SuS ihre Kenntnisse nach Möglichkeit in einem größeren Projekt anwenden und verknüpfen.

### 3.1.2.2. Leistungsaspekte

#### Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen

#### Praktische Leistungen am Computer

- Modellierung, Implementierung, Test und Anwendung von Informatiksystemen

#### Sonstige schriftliche Leistungen

- Lernerfolgsüberprüfung durch kurze schriftliche Übungen  
Über die Durchführung schriftlicher Übungen entscheidet die Lehrkraft.  
Schriftliche Übung dauern ca. 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- Bearbeitung von schriftlichen Aufgaben im Unterricht

### 3.1.2.3. Kriterien

Die folgenden allgemeinen Kriterien gelten sowohl für die mündlichen als auch für die schriftlichen Formen der sonstigen Mitarbeit.

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,

- 
- die Reflexion des eigenen Handelns und
  - die Aufnahme von Beratung durch die Lehrkraft sowie durch Mitschülerinnen und Mitschüler.

#### **3.1.2.4. Grundsätze der Leistungsrückmeldung und Beratung**

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch ein Gespräch mit den Eltern

erfolgen.

Leistungsrückmeldungen erfolgen in der Einführungsphase auch im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.

## **4. Entscheidungen zu fach- und unterrichtsübergreifenden Fragen**

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

### **4.1. Zusammenarbeit mit anderen Fächern**

In der Gymnasialen Oberstufe werden Projektkurse angeboten. Sofern es die zur Verfügung stehenden Lehrerstunden im Fach Informatik erlauben, können Projektkurse mit dem Leitfach Informatik angeboten werden. Mögliche Themenfestlegungen trifft die ausrichtende Lehrkraft in Absprache mit Fachvertretern anderer Bezugsfächer.

### **4.2. 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 in der Regel nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

---

## 5. 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.

Am Ende jedes Schuljahres findet unter den Fachkollegen ein Austausch über das abgelaufene Schuljahr mit Blick auf das schulinterne Curriculum statt. Sollten sich dabei Änderungsbedarfe ergeben, werden diese mit dem Emsland Gymnasium abgesprochen.

Nach Abschluss des Abiturs wird die Fachkonferenz Informatik auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und nach Rücksprache mit dem Emsland Gymnasium ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.

Der Unterricht selbst sollte in regelmäßigen Abständen mit SEFU-online oder den in der Lehrerkonferenz abgesprochenen Werkzeugen evaluiert werden.