



Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten gleichwohl für beiderlei Geschlecht.

verabschiedet auf der Fachkonferenz Informatik am 10.06.2026

Curriculum im Fach Informatik für die SII

Inhalt

1	Rahmenbedingungen der fachlichen Arbeit	2
2	Entscheidungen zum Unterricht	3
2.1.	Unterrichtsvorhaben	3
2.2.	Übersichtsraster Unterrichtsvorhaben (GK und LK)	4
2.3.	Konkretisierte Unterrichtsvorhaben (GK und LK)	7
3	Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	26
3.1.	Zusammenarbeit mit anderen Fächern	26
3.2.	Vorbereitung auf die Erstellung der Facharbeit	26
3.3.	Exkursionen / Workshops	26
3.4.	Evaluation des schulinternen Curriculums	26
3.5.	Fachgruppenarbeit	27
4	Grundsätze der Leistungsbewertung und -rückmeldung	27
4.1.	Rechtliche Grundlagen und Prinzipien	27
4.2.	Anforderungen und Kriterien zur Beurteilung (EF – Q2)	28
	<i>Klausuren</i>	28
	<i>Sonstige Mitarbeit</i>	29
4.3.	Individuelle Förderung	32
4.4.	Bildung der Zeugnisnote	32

1 Rahmenbedingungen der fachlichen Arbeit

Mit Blick auf eine als heterogen zu bezeichnende Zusammensetzung unserer Schülerschaft besteht ein wesentliches Leitziel der Schule in der individuellen Förderung. Die Fachgruppe Informatik versucht in besonderem Maße, jeden Lernenden in seiner Kompetenzentwicklung möglichst weit zu bringen. Außerdem wird angestrebt, Interesse an einem Studium oder Beruf mit informatischem Bezug zu wecken. In diesem Rahmen sollen u.a. Schüler mit besonderen Stärken im Bereich Informatik unterstützt werden. Dieses drückt sich in AG-Angeboten (Robotic-AG, Web-AG, etc.) ebenso aus wie in der regelmäßigen Teilnahme von Schülergruppen an Wettbewerben wie Informatik-Biber, Jugendwettbewerb Informatik, Jugend forscht oder dem Bundeswettbewerb Informatik. In enger Kooperation mit der RWTH Aachen ermöglichen wir besonders begabten Lernenden die Teilnahme an Seminaren und Praktika. Hier können sie sogar schon Leistungsnachweise erwerben, die ihnen in einem späteren Studium anerkannt werden.

Nicht nur in der SI, sondern auch in der Oberstufe ist der Austausch unter den Fachkollegen zu Inhalten, methodischen Herangehensweisen und zu fachdidaktischen Problemen intensiv. So ist u.a. eine Intensivierung kollegiumsinterner Fortbildungen geplant.

Die Ausstattung mit geeigneten Fachräumen, Computern und mit Materialien wie Platinen, Roboter-Bausätzen o.ä. ist vielseitig und umfangreich. Der Etat für Neuanschaffungen und Reparaturen ist nicht üppig, aber ausreichend, jedoch sind aufwendige Neuanschaffungen über diesen Etat oft nicht zu stemmen, sondern müssen anderweitig finanziert werden. Schrittweise sollen mehr Möglichkeiten für schüleraktivierenden Unterricht an geeigneten Stellen geschaffen werden, was nicht nur Technik, sondern auch die Beschäftigung mit informatischen Inhalten ohne die Nutzung von Computern erlaubt. Darüber hinaus setzen wir Schwerpunkte in der Nutzung von neuen Medien, wozu regelmäßig kollegiumsinterne Fortbildungen angeboten werden. Dazu gehört auch die Erfassung von Daten und Messwerten sowie die Datenverarbeitung und -verwaltung.

In der Oberstufe ist das Fach Informatik in der Regel in der Einführungsphase mit Grundkursen, in der Qualifikationsphase je Jahrgangsstufe mit einem Grundkurs vertreten. Leistungskurse kommen je nach Wahlverhalten der Schüler als eigenständiger Leistungskurs oder als sogenannter „Huckepackkurs“ zustande, in denen ein Teil der Unterrichtsstunden zusammen mit einem Grundkurs unterrichtet wird. Die Lehrbesetzung in Informatik ermöglicht einen ordnungsgemäßen Fachunterricht in beiden Sekundarstufen. Für den Informatikunterricht in der Sekundarstufe II ist an der Schule derzeit das Schulbuch Informatik (Schöningh Verlag) eingeführt.

Huckepackmodell

Im Rahmen des sogenannten Huckepackmodells werden die Schüler des Leistungskurses gemeinsam mit dem Grundkurs unterrichtet. Die Schüler des Leistungskurses bekommen in den Arbeitsphasen des Unterrichts Aufgaben auf höherem Anspruchsniveau als die Schüler des Grundkurses. Da die Kurse relativ klein sind (ansonsten würde es getrennte Kurse geben), stellt die Binnendifferenzierung kein Problem dar. Die zusätzlich vorgesehenen zwei Wochenstunden des Leistungskurses dienen einerseits der Behandlung der in den jeweiligen Unterrichtsvorhaben ausgewiesenen leistungskurs-spezifischen Inhalte und Kompetenzen (in den Unterrichtsvorhaben **rot** markiert). Diese werden in den gesonderten LK-Stunden vertiefend und in der für den Leistungskurs erforderlichen fachlichen Breite und Tiefe erarbeitet.

Andererseits werden die zusätzlichen Stunden genutzt, um die gemeinsam mit dem Grundkurs behandelten Inhalte zu festigen, zu erweitern und zu vertiefen. Dies kann insbesondere durch weiterführende fachliche Analysen, komplexere Problemstellungen, zusätzliche praktische Umsetzungen, projektorientierte Arbeitsformen sowie eine intensivere Reflexion informatischer Konzepte erfolgen. Auf diese Weise wird sichergestellt, dass die Anforderungen des Leistungskurses gemäß den Vorgaben für die gymnasiale Oberstufe erfüllt werden und zugleich eine angemessene individuelle Förderung der Schüler stattfindet.

Für die Schüler des Grundkurses entstehen durch dieses Organisationsmodell keine Nachteile. Die im Kernlehrplan für den Grundkurs vorgesehenen Inhalte, Kompetenzerwartungen und Unterrichtszeiten werden vollständig umgesetzt. Die gemeinsame Unterrichtsphase orientiert sich an den Anforderungen des Grundkurses; leistungskursspezifische Erweiterungen erfolgen ausschließlich in den zusätzlichen LK-Stunden. Somit wird gewährleistet, dass die Lernenden des Grundkurses alle für ihren Bildungsgang vorgesehenen fachlichen Inhalte in vollem Umfang bearbeiten, während die Schüler des Leistungskurses die erforderlichen zusätzlichen Vertiefungen und Erweiterungen erhalten. Darüber hinaus sind die im Leistungskurs erarbeiteten Inhalte für die Schüler des Grundkurses über ein gemeinsames digitales Kursnotizbuch jederzeit transparent und zugänglich. Die Leistungskursschüler berichten zudem in regelmäßigen Abständen im gemeinsamen Unterricht über ihre Projekte, Vorhaben, Arbeitsergebnisse und gewonnenen Erkenntnisse. Auf diese Weise erhalten auch die Lernenden des Grundkurses Einblicke in weiterführende informatische Fragestellungen und Arbeitsweisen, ohne dass daraus zusätzliche verbindliche Anforderungen für sie erwachsen. Die gemeinsame Kursstruktur fördert damit den fachlichen Austausch zwischen Grund- und Leistungskurs und trägt zu einer erweiterten Lernkultur innerhalb des Faches Informatik bei.

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, Lerngelegenheiten für die Lerngruppe so anzulegen, dass alle Kompetenzerwartungen des Kernlehrplans von den Schülern erworben werden können. Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ wird die für alle Lehrer gemäß Fachkonferenzbeschluss **verbindliche Verteilung der Unterrichtsvorhaben** dargestellt. Das Übersichtsraster dient dazu, den Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten sowie in der Fachkonferenz verabredeten verbindlichen Kontexten zu verschaffen. Um Klarheit für die Lehrkräfte herzustellen und die Übersichtlichkeit zu gewährleisten, werden in der Kategorie „Kompetenzen“ an dieser Stelle nur die übergeordneten Kompetenzerwartungen ausgewiesen, während die konkretisierten Kompetenzerwartungen erst auf der Ebene konkretisierter Unter-



richtsvorhaben Berücksichtigung finden. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Spielraum 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. 75-80 Prozent der Bruttounterrichtszeit geplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ einschließlich der dort genannten Kontexte zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, besitzt die **exemplarische Ausweisung konkretisierter Unterrichtsvorhaben empfehlenden Charakter**, es sei denn, die Verbindlichkeit bestimmter Aspekte ist dort, markiert durch Fettdruck, explizit angegeben. Insbesondere Referendaren sowie neuen Kollegen dienen die konkretisierten Unterrichtsvorhaben vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen. Abweichungen von den empfohlenen Vorgehensweisen bezüglich der konkretisierten Unterrichtsvorhaben sind im Rahmen der pädagogischen Freiheit der Lehrkräfte jederzeit möglich. Sicherzustellen bleibt allerdings auch hier, dass im Rahmen der Umsetzung der Unterrichtsvorhaben insgesamt alle Kompetenzerwartungen des Kernlehrplans Berücksichtigung finden (UF: Umgang mit Fachwissen, E: Erkenntnisgewinn, K: Kommunikation, B: Bewertung).

2.2. Übersichtsraster Unterrichtsvorhaben (GK und LK)

Unterrichtsvorhaben EF-I Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik Kompetenzen: Argumentieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren Inhaltsfelder: Informatiksysteme - Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Einzelrechner, Dateisystem, Internet • Einsatz, Nutzung und Aufbau von Informatiksystemen • Wirkung der Automatisierung • Eigenschaften digitaler Datenspeicherung und -verarbeitung	Unterrichtsvorhaben EF-II Grundlagen der objektorientierten Analyse und Programmierung an Beispielen Kompetenzen: Modellieren, Darstellen und Interpretieren, Implementieren, Kommunizieren und Kooperieren Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf u. Implementierung einf. Algorithmen • Syntax und Semantik einer Programmiersprache • Binäre Codierung und Verarbeitung
---	---



Unterrichtsvorhaben EF-III Objektorientierte Analyse, Modellierung und Implementierung von Klassen- und Objektbeziehungen Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen u. Interpretieren, Kommunizieren und Kooperieren Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf u. Implementierung einf. Algorithmen • Syntax und Semantik einer Programmiersprache • Dateisystem (Rechteverwaltung), Internet	[Unterrichtsvorhaben EF-IV] Implementierung und Entwurf graphischer Benutzeroberflächen (GUI) Kompetenzen: Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf u. Implementierung einf. Algorithmen • Syntax und Semantik einer Programmiersprache
Unterrichtsvorhaben EF-V Verwaltung großer Datenmengen – Such- und Sortieralgorithmen Kompetenzen: Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren Inhaltsfelder: Daten und ihre Strukturierung – Algorithmen (- Informatiksysteme) Inhaltliche Schwerpunkte: • Objekte und Klassen • Algorithmen zum Suchen und Sortieren • Analyse, Entwurf u. Implementierung einf. Algorithmen • Dateisystem • (Einsatz von Informatiksystemen • Wirkung der Automatisierung)	Unterrichtsvorhaben EF-VI Grundlagen der digitalen Datenverarbeitung Kompetenzen: Argumentieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren Inhaltsfelder: (Formale Sprachen und Automaten -) Informatiksysteme - Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • (Syntax und Semantik einer Programmiersprache) • Digitalisierung • (Einzelrechner) • Wirkung der Automatisierung • Geschichte der automatischen Datenverarbeitung
Unterrichtsvorhaben Q1-I Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen LK: Vertiefung an ausgewählten Beispielen Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen u. Interpretieren, Kommunizieren u. Kooperieren Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme Inhaltliche Schwerpunkte: • Objekte und Klassen • Analyse, Entwurf u. Implementierung von Algorithmen • Algorithmen in ausgewählten Kontexten • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen	Unterrichtsvorhaben Q1-II Rekursive und iterative Algorithmen im Anwendungskontext Sortieren und Suchen LK: Maschinennahe Programmierung und Grenzen der Berechenbarkeit Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen u. Interpretieren, Kommunizieren u. Kooperieren Inhaltsfelder: Algorithmen - Formale Sprachen und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: • Analyse, Entwurf u. Implementierung von Algorithmen • Algorithmen in ausgewählten Kontexten • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen • Grenzen der Automatisierung • LK: Einzelrechner und Rechnernetzwerke, Grenzen und Wirkung der Automatisierung



Unterrichtsvorhaben Q1-III

Endliche Automaten und Formale Sprachen, LK: Compiler und formale Sprachen

Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen u. Interpretieren, Kommunizieren u. Kooperieren

Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprache und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft – **Kellerautomaten u. kontextfreie Sprachen**

Inhaltliche Schwerpunkte:

- Endliche Automaten, **LK: und Kellerautomaten**
- Grammatiken regulärer Sprachen, **LK: und kontextfreie Sprachen**
- **Scanner, Parser und Interpreter für eine reguläre Sprache**
- Möglichkeiten u. Grenzen von Automaten u. formalen Sprachen
- Nutzen von Informatiksystemen

Unterrichtsvorhaben Q2-I

Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume

LK: ... und am Beispiel der Graphen

Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren

Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme

Inhaltliche Schwerpunkte:

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

Unterrichtsvorhaben Q2-II

Kommunikation, Sicherheit und Datenschutz in Netzstrukturen

LK: Modellierung und Implementierung von Client-Server-Anwendungen

Kompetenzen: Argumentieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren, **Modellieren, Implementieren**

Inhaltsfelder: Informatiksysteme - Informatik, Mensch und Gesellschaft, **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**
- Einzelrechner und Rechnernetzwerke
- Sicherheit
- Nutzung von Informatiksystemen
- Wirkungen der Automatisierung

Unterrichtsvorhaben Q2-III

Modellierung und Nutzung relationaler Datenbanken, LK: .. und Implementierung

Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren

Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprache und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Datenbanken
- Algorithmen in ausgewählten informatischen Kontexten
- Syntax und Semantik einer Programmiersprache
- Nutzung von Informatiksystemen
- Sicherheit
- Wirkung der Automatisierung

2.3. Konkretisierte Unterrichtsvorhaben (GK und LK)

EF-I: Grundlagen der objektorientierten Analyse und Programmierung an Beispielen Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten
Leitfrage: 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? Inhaltliche Schwerpunkte: • Objekte und Klassen, Analyse, Entwurf u. Implementierung einf. Algorithmen, Syntax und Semantik einer Programmiersprache, Binäre Codierung und Verarbeitung Zeitbedarf: 6 Stunden	

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Grundlegende Konzepte der objektorientierten Programmierung a) Sichtweise der objektorientierten Informatik auf die Welt b) OOP als informatikspezifische Modellierung der Realität c) Schritte der Softwareentwicklung	Die Schüler ... - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - analysieren und erläutern objektorientierte Modellierungen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M),	Kapitel 1 Konzepte des objektorientierten Modellierens 1.1 Modellierung der Realität 1.2 Die Welt ist voller Objekte Projekteinstieg: Klassenentwurf – step by step

2. Erweiterung der objektorientierten Programmierung a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm b) Objektkommunikation im Sequenzdiagramm c) Klassendokumentation d) Umsetzung von Teilen der Modellierung	- modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).	1.3 Gut geplant – Klassenentwurf 1.4 Hierarchien machen's einfacher – Vererbung
3. Mensch und Technik a) Informatiker verändern die Welt b) Automatisierung des Alltags durch Informatik		Die digitale Welt 001 – Mensch und Technik

EF-IIa: Grundlagen der objektorientierten Analyse und Programmierung an Beispielen	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten
Leitfrage: Wie lassen sich Gegenstandsbereiche informatisch modellieren und in einem Greenfoot-Szenario informatisch realisieren? Inhaltliche Schwerpunkte: • Objekte und Klassen, Analyse, Entwurf u. Implementierung einf. Algorithmen, Syntax und Semantik einer Programmiersprache, Binäre Codierung und Verarbeitung Zeitbedarf: 6 Stunden	

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 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 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 5) 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Identifikation von Objekten und Klassen (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 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 Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).	Kapitel 2 Einführung in die Objektorientierung 2.1 Objektorientierte Modellierung
2. Analyse von Objekten und Klassen im Greenfoot-Szenario (a) Schritte der objektorientierten Analyse, Modellierung und Implementation (b) Analyse und Erprobung der Objekte im Greenfoot-Szenario		Kapitel 2 Einführung in die Objektorientierung 2.2 Das Greenfoot-Szenario „Planetenerkundung“ Von der Realität zu Objekten Von den Objekten zu Klassen Klassendokumentation Objekte inspizieren Methoden aufrufen Objektidentität und Objektzustand
3. Implementierung einfacher Aktionen in Greenfoot (a) Quelltext einer Java-Klasse (b) Implementation eigener Methoden, Dokumentation mit Javadoc (c) Programme übersetzen (Aufgabe des Compilers) und testen		Kapitel 2 Einführung in die Objektorientierung 2.3 Programmierung in Greenfoot Methoden schreiben Programme übersetzen und testen

EF-Ilb: Algorithmische Grundstrukturen in Java

Inhaltsfeld: Informatiksysteme; Informatik, Mensch und Gesellschaft

Leitfrage: Wie lassen sich Aktionen von Objekten flexibel realisieren?

Inhaltliche Schwerpunkte: • Einzelrechner, Dateisystem, Internet; • Einsatz, Nutzung und Aufbau von Informatiksystemen; • Wirkung der Automatisierung; • Eigenschaften digitaler Datenspeicherung und -verarbeitung

Zeitbedarf: 6 Stunden

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 Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (a) Wiederholungen (While-Schleife, Zählschleife) (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 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).	Kapitel 3 Algorithmen 3.1 Kontrollstrukturen 3.2 Wiederholungen 3.3 Zählschleifen 3.4 Bedingte Anweisungen 3.5 Logische Operationen 3.6 Algorithmen entwickeln
2. Variablen und Methoden (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		Kapitel 4 Variablen und Methoden 4.1 Variablen 4.2 Methoden

EF-IIc: Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung	Inhaltsfeld: Informatiksysteme; Informatik, Mensch und Gesellschaft
Leitfrage: 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?	
Inhaltliche Schwerpunkte: • Einzelrechner, Dateisystem, Internet; • Einsatz, Nutzung und Aufbau von Informatiksystemen; • Wirkung der Automatisierung; • Eigenschaften digitaler Datenspeicherung und -verarbeitung	
Zeitbedarf: 2 Stunden	

Vorhabenbezogene Konkretisierung:

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

Im ersten Schritt erarbeiten die 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 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Analoge und digitale Aufbereitung und Verarbeitung von Daten (a) Erarbeitung der Unterschiede von analog und digital (b) Zusammenfassung und Bewertung der technischen Möglichkeiten von analog und digital	Die 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) - 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)	<i>Die digitale Welt 001 – Von analog zu digital</i>
2. Der Umgang mit binärer Codierung von Informationen (a) Das binäre (und hexadezimale) Zahlensystem (b) Binäre Informationsspeicherung (c) Binäre Verschlüsselung (d) Implementation eines Binärumrechners		<i>Die digitale Welt 010 – Binäre Welt</i>
3. Aufbau informatischer Systeme (a) Identifikation des EVA-Prinzips als grundlegende Arbeitsweise informatischer Systemen (b) Nachvollziehen der von-Neumann-Architektur als relevantes Modell der Umsetzung des EVA-Prinzips		<i>Die digitale Welt 011 – Der Von-Neumann-Rechner</i>

[EF-IV: Implementierung und Entwurf graphischer Benutzeroberflächen (GUI)]	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten
Inhaltliche Schwerpunkte: • Objekte und Klassen, Analyse, Entwurf u. Implementierung einf. Algorithmen, Syntax und Semantik einer Programmiersprache Zeitbedarf: 10 Stunden	

optional

EF-III: Objektorientierte Analyse, Modellierung und Implementierung von Klassen- und Objektbeziehungen	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme
Leitfrage: Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert? Inhaltliche Schwerpunkte: • Objekte und Klassen, Analyse, Entwurf u. Implementierung einf. Algorithmen, Syntax und Semantik einer Programmiersprache, Dateisystem (Rechteverwaltung), Internet Zeitbedarf: 14 Stunden	

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 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 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	Die Schüler - analysieren und erläutern eine objektorientierte Modellierung (A), - stellen die Kommunikation zwischen Objekten grafisch dar (M), - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), - modellieren Klassen unter Verwendung von Vererbung (M), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - analysieren und erläutern einfache Algorithmen und Programme (A) - modifizieren einfache Algorithmen und Programme (I), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M). - stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D)	Kapitel 5 Klassenentwurf 5.1 Von der Realität zum Programm 5.2 Objekte 5.3 Klassen und Beziehungen entwerfen
2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden (b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung		Kapitel 5 Klassenentwurf 5.4 Klassen und Beziehungen implementieren
3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes (a) Klassen werden in Java-Quellcode umgesetzt. (b) Das Geheimnisprinzip wird umgesetzt. (c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.		Kapitel 5 Klassenentwurf 5.4 Klassen und Beziehungen implementieren
4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet. (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert		Kapitel 5 Klassenentwurf 5.5 Vererbung
5. Softwareprojekt		Kapitel 7 Softwareprojekte

(a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) (b) Erweiterung des Spiels um weitere Funktionalitäten (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche (d) (arbeitsteilige) Implementation des Spiels	7.1 Softwareentwicklung 7.2 Oberflächen
---	--

[EF-IV: Implementierung und Entwurf graphischer Benutzeroberflächen (GUI)]	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten
Inhaltliche Schwerpunkte: • Objekte und Klassen, Analyse, Entwurf u. Implementierung einf. Algorithmen, Syntax und Semantik einer Programmiersprache Zeitbedarf: 10 Stunden	

optional

EF-V: Verwaltung großer Datenmengen – Such- und Sortialgorithmen	Inhaltsfelder: Daten und ihre Strukturierung – Algorithmen (- Informatiksysteme)
Leitfrage: Wie können Objekte bzw. Daten effizient gesucht und sortiert werden? Inhaltliche Schwerpunkte: • Objekte und Klassen, Algorithmen zum Suchen und Sortieren, Analyse, Entwurf u. Implementierung einf. Algorithmen, Dateisystem, (Einsatz von Informatiksystemen, Wirkung der Automatisierung) Zeitbedarf: 16 Stunden	

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortialgorithmen. 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 Schüler das Feld als eine erste Datensammlung kennen. Alle Algorithmen dieses Kapitels arbeiten auf einem Feld. Die Schüler lernen zunächst Strategien des Suchens (lineare Suche, binäre Suche, Hashing) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Daran anschließend lernen die 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 Sortierv Verfahren beurteilt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Modellierung und Implementation von Datenansammlungen (a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld	Die Schüler - analysieren Such- und Sortialgorithmen und wenden sie auf Beispiele an (D) - entwerfen einen weiteren Algorithmus zum Sortieren (M)	Kapitel 6 Suchen und Sortieren 6.1 Speichern mit Struktur – Arrays

2. Explorative Erarbeitung von Suchverfahren (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	- beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A) - ordnen Attributen lineare Datenansammlungen zu (M)	Kapitel 6 Suchen und Sortieren Projekteinstieg 1: Suchen 6.2 Suchen mit System Lineare Suche Binäre Suche Hashing
3. Systematisierung von Algorithmen und Effizienzbetrachtungen (a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)		Kapitel 6 Suchen und Sortieren Projekteinstieg 2: Sortieren 6.3 Ordnung ist das halbe Leben!? – Sortieren Sortieren Selection Sort Insertion Sort Bubble Sort

EF-VI: Grundlagen der digitalen Datenverarbeitung <i>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</i>	Inhaltsfelder: (Formale Sprachen und Automaten -) Informatiksysteme - Informatik, Mensch und Gesellschaft
Leitfrage: Welche Entwicklungen, Ideen und Erfindungen haben zur heutigen Informatik geführt? Welche Auswirkungen hat die Informatik für das Leben des modernen Menschen? Inhaltliche Schwerpunkte: • (Syntax und Semantik einer Programmiersprache), Digitalisierung, (Einzelrechner), Wirkung der Automatisierung, Geschichte der automatischen Datenverarbeitung Zeitbedarf: 4 Stunden	

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.

Unterrichtssequenzen 1. Schriftzeichen, Rechenmaschine, Computer (a) Anhand von Schwerpunkten, wie z.B. Datenspeicherung, Maschinen, Vernetzung sollen wichtige Entwicklungen der	Zu entwickelnde Kompetenzen Die Schüler - bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),	Kapitel und Materialien <i>Die digitale Welt 100 – Von der Schrift zum Smartphone</i>
---	--	---

Informatik vorgestellt werden. (b) Anhand der unterschiedlichen Schwerpunkte sollen universelle Tendenzen der Entwicklung der Informationsverarbeitung erarbeitet werden.	- erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) - nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)	
2. Die Informationsverarbeitung und ihre Möglichkeiten und Gefahren (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.		<i>Die digitale Welt 100 – Das Leben in der digitalen Welt</i>

Q1-Ia: Wiederholung und Vertiefung der objektorientierten Modellierung	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme
Leitfrage: 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? Inhaltliche Schwerpunkte: • Objekte und Klassen, • Analyse, Entwurf und Implementierung von Algorithmen, • Algorithmen in ausgewählten informatischen Kontexten, • Syntax und Semantik einer Programmiersprache, Nutzung von Informatiksystemen Zeitbedarf: 4 Stunden	

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 Objektidentifikation 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 Schüler Klassenentwürfe.

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
-----------------------------	------------------------------------	--------------------------------

1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung a) Sichtweise der objektorientierten Informatik auf die Welt b) OOP als informatikspezifische Modellierung der Realität c) Schritte der Softwareentwicklung	Die Schüler ... - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - analysieren und erläutern objektorientierte Modellierungen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).	Kapitel 1 Konzepte des objektorientierten Modellierens 1.1 Modellierung der Realität 1.2 Die Welt ist voller Objekte Projekteinstieg: Klassenentwurf – step by step 1.3 Gut geplant – Klassenentwurf 1.4 Hierarchien machen's einfacher – Vererbung
2. Erweiterung der objektorientierten Programmierung a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm b) Objektkommunikation im Sequenzdiagramm c) Klassendokumentation d) Umsetzung von Teilen der Modellierung		
3. Mensch und Technik a) Informatiker verändern die Welt b) Automatisierung des Alltags durch Informatik		Die digitale Welt 001 – Mensch und Technik

Q1-Ib: Modellierung und Implementierung dynamischer Listenstrukturen und deren Anwendungen; Vertiefung an ausgewählten Beispielen	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme
Leitfrage: Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können? Inhaltliche Schwerpunkte: • Objekte und Klassen, • Analyse, Entwurf und Implementierung von Algorithmen, • Algorithmen in ausgewählten informatischen Kontexten, • Syntax und Semantik einer Programmiersprache, Nutzung von Informatiksystemen Zeitbedarf: 16 Stunden (LK 33 Stunden)	

Vorhabensbezogene Konkretisierung:

Ausgehend von 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Die Datenstruktur Feld a) Erarbeitung der Anforderungen an eine Datenstruktur b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays c) Modellierung und Implementierung von Anwendungen	Die Schüler ... - erläutern Operationen dynamischer (linearer) Datenstrukturen (A) - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D) - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M) - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) - dokumentieren Klassen (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) - verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M) - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).	Kapitel 2 Lineare Datenstrukturen 2.1 Anforderungen an eine Datenstruktur 2.2 Datenansammlungen fester Größe – Arrays
2. Die Datenstruktur Schlange a) Modellierung und Implementierung der Verknüpfung von Objekten b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS. c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i> ; Erschließen der Standardoperationen e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i>		2.3 Wer zuerst kommt ... – Schlangen
3. Die Datenstruktur Stapel a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden b) Funktionalität der Klasse <i>Stapel</i> unter Verwendung der Klasse <i>Stack</i> , Erschließen der Standardoperationen c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i> d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i> , <i>Stack</i> und <i>Array (Palindrom)</i>)		2.4 Daten gut abgelegt – Stapel

4. Die Datenstruktur Liste 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>) b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i> c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i> d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List)		2.5 Flexibel für alle Fälle – lineare Listen
5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen		2.6 Prüfungsvorbereitung
6. Datenschutz a) Datenschutz als Grundrecht b) Das Datenschutzgesetz c) Datensammler		<i>Projekteinstieg: Wartende Helden</i> <i>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.</i> Die digitale Welt 101 – Datenschutz

Q1-II: Rekursive und iterative Algorithmen im Anwendungskontext Sortieren und Suchen <i>Maschinennahe Programmierung und Grenzen der Berechenbarkeit</i>	Inhaltsfelder: Algorithmen - Formale Sprachen und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft
---	---

Leitfrage: 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?

Inhaltliche Schwerpunkte: • Analyse, Entwurf und Implementierung von Algorithmen, Algorithmen in ausgewählten Kontexten, Syntax und Semantik einer Programmiersprache, Nutzung von Informatiksystemen, Grenzen der Automatisierung, **LK: Einzelrechner und Rechnernetzwerke, Grenzen und Wirkung der Automatisierung**

Zeitbedarf: 30 Stunden (**LK 50 Stunden**)

Vorhabenbezogene Konkretisierung:

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmusentwicklung 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 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 wird.

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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Eigenschaften von Algorithmen a) Qualitätseigenschaften von Algorithmen b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“ c) Analyse und Entwicklung von rekursiven Algorithmen	Die Schüler ... - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), - implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - testen Programme systematisch anhand von Beispielen (I), - implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), - beurteilen die syntaktische Korrektheit und die Funktionalität von	Kapitel 3 Algorithmen 3.1 Ohne Algorithmen läuft nichts 3.2 Teile die Arbeit – rekursive Algorithmen
2. Suchen in Listen und Arrays a) Lineare Suche in Listen und Arrays b) Binäre Suche in einem Array c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf		3.3 Suchen – iterativ und rekursiv Projekteinstieg: Laufzeitanalyse experimentell

3. Sortieren auf Listen und Arrays a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (Sortieren durch Einfügen) b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für eine Liste (Quicksort) c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf d) Weitere Sortierverfahren auf Listen und Arrays (Sortieren durch Auswählen, Mergesort)	Programmen (A), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A).	3.4 Sortieren – iterativ und rekursiv
4. Verantwortung der Informatik a) Der Unterschied zwischen Anwender und Produzent von Informatiksystemen b) Informatik und Ethik		Die digitale Welt 011 – Verantwortung der Informatik

Q1-III: Endliche Automaten und Formale Sprachen <i>Compiler und formale Sprachen</i>	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprache und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft – Kellerautomaten u. kontextfreie Sprachen
Leitfrage: 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? Was sind die strukturellen Hauptbestandteile eines Computers und wie kann man sich die Ausführung eines maschinennahen Programms mit diesen Komponenten vorstellen? Welche Möglichkeiten bieten Informatiksysteme und wo liegen ihre Grenzen? Inhaltliche Schwerpunkte: • Endliche Automaten und Kellerautomaten, Grammatiken regulärer Sprachen und kontextfreie Sprachen, Scanner, Parser und Interpreter für eine reguläre Sprache, Möglichkeiten u. Grenzen von Automaten u. formalen Sprachen, Nutzen von Informatiksystemen Zeitbedarf: 25 Stunden (LK 42 Stunden)	

Vorhabensbezogene 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Endliche Automaten a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus c) Analyse der Eigenschaften von DEAs durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere	Die Schüler ... - analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A), - ermitteln die Sprache, die ein endlicher Automat akzeptiert (D), - entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), - stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), - entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), - analysieren und erläutern Grammatiken regulärer Sprachen (A), - modifizieren Grammatiken regulärer Sprachen (M), - ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), - entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M) - entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M), - beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D), - zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), - erläutern die Ausführung eines einfachen maschinennahen Programms sowie die Datenspeicherung auf einer „Von-Neumann-Architektur“ (A), - untersuchen und beurteilen Grenzen des Problemlösens mit Informatiksystemen (A).	Kapitel 4 Endliche Automaten und formale Sprachen 4.1 Endliche Automaten Projekteinstieg: Schatzsuche
2. Grammatiken regulärer Sprachen a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften) b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.		4.2 Formale Sprachen
3. Übungen und Vertiefungen Verwendung endlicher Automaten und Grammatiken regulärer Sprachen		4.5 Prüfungsvorbereitung
4. Grundsätzliche Arbeitsweise eines Computers und Grenzen der Berechenbarkeit a) Von-Neumann-Architektur und die Ausführung maschinennaher Programme b) Grenzen der Berechenbarkeit anhand des Halteproblems, nicht effizient berechenbare Probleme		Die digitale Welt 101 – Maschinennahe Programmierung Die digitale Welt 100 – Berechenbarkeit

Q2-I: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der

Inhaltsfeld: Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatiksysteme

Binärbäume und am Beispiel der Graphen

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

Inhaltliche Schwerpunkte: • Objekte und Klassen, Analyse, Entwurf und Implementierung von Algorithmen, Algorithmen in ausgewählten informatischen Kontexten, Syntax und Semantik einer Programmiersprache, Nutzung von Informatiksystemen

Zeitbedarf: 20 Stunden (LK 33 Stunden)

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, 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 *ComparableContent* 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 *ComparableContent* modelliert und (ggf. in Teilen) implementiert.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Aufbau von Baumstrukturen und Grundbegriffe a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten	Die Schüler ... - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), - analysieren und erläutern Algorithmen und Programme (A), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	Kapitel 5 Nicht-lineare Datenstrukturen 5.1 Spielen mit Struktur – Baumstrukturen Projekteinstieg 1: Spielbäume
2. Binäre Bäume a) rekursiver Aufbau eines binären Baums b) Traversierungen (pre-, in-, postorder) c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse <i>BinaryTree</i> (als Entwurfs- und Implementationsdiagramm) d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)	- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare	5.2 Zwei Nachfolger sind genug! - Binäre Bäume Implementation des Projekts Ahnenbaum

3. Binäre Suchbäume a) Prinzip des binären Suchbaums, Ordnungsrelation b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe) c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)	Datensammlungen zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), - implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - modifizieren Algorithmen und Programme (I), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I).	5.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume Projekteinstieg 2: Binäre Suchbäume Implementation des Projekts Benutzerverwaltung
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		5.8 Prüfungsvorbereitung

Q2-II: Kommunikation, Sicherheit und Datenschutz in Netzstrukturen LK: Modellierung und Implementierung von Client-Server-Anwendungen	Inhaltsfelder: Informatiksysteme - Informatik, Mensch und Gesellschaft, Daten und ihre Strukturierung, Algorithmen, Formale Sprachen und Automaten
Leitfrage: 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? Inhaltliche Schwerpunkte: Objekte und Klassen, Analyse, Entwurf und Implementierung von Algorithmen, Algorithmen in ausgewählten informatischen Kontexten, Syntax und Semantik einer Programmiersprache, Einzelrechner und Rechnernetzwerke, Sicherheit, Nutzung von Informatiksystemen, Wirkungen der Automatisierung Zeitbedarf: 16 Stunden (LK 28 Stunden)	

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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Technische Kommunikation als Fortführung natürlicher	Die Schüler ...	Kapitel 6 Kommunikation in

Kommunikation a) Kommunikation im Shannon-Weaver-Modell b) Kriterien von technischen Kommunikationsarten c) Die Geschichte der technischen Kommunikation	- beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zur Erschließung, Aufbereitung und Präsentation fachlicher Inhalte (D).	Netzwerken 6.1 Menschen kommunizieren – ohne und mit Technik Projekteinstieg: Kommunikation im Wilden Westen
2. Aufbau von Netzwerken und Kommunikationsregeln a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung b) Geregelte technische Kommunikation durch Protokolle in Schichtenmodellen		6.2 Ohne Protokoll läuft nichts – Netzwerke
3. Aufgabenteilung in Netzwerken durch Server und Client a) Aufbau und Aufgaben der Client-Server-Struktur b) Protokolle zwischen Client und Server		6.3 Einer für alle – Client-Server-Struktur
4. Kryptologie a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA) b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand		Die digitale Welt 111 – Kryptologie
5. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netzwerken		6.4 Prüfungsvorbereitung

Q2-III: Modellierung und Nutzung relationaler Datenbanken und Implementierung	Inhaltsfelder: Daten und ihre Strukturierung - Algorithmen - Formale Sprache und Automaten - Informatiksysteme - Informatik, Mensch und Gesellschaft
Leitfrage: Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext? Inhaltliche Schwerpunkte: • Datenbanken, • Algorithmen in ausgewählten informatischen Kontexten, • Syntax und Semantik einer Programmiersprache, • Nutzung von Informatiksystemen, • Sicherheit, Wirkung der Automatisierung Zeitbedarf: 20 Stunden (LK 33 Stunden)	

Vorhabenbezogene Konkretisierung:

Am Beispiel eines Online-Buchhandels wird der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schüler nehmen dabei zunächst die Sicht der Anwender ein, 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 (soweit nötig) normalisiert.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen b) SQL-Abfragen - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) - Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke) c) Vertiefung an einem weiteren Datenbankbeispiel	Die Schüler ... - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), - modifizieren eine Datenbankmodellierung (M), - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - bestimmen Primär- und Sekundärschlüssel (M), - analysieren und erläutern eine Datenbankmodellierung (A), - erläutern die Eigenschaften normalisierter Datenbankschemata (A), - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). - überführen Datenbankschemata in die 1. bis 3. Normalform (M).	Kapitel 8 Datenbanken 7.1 Wissen speichern und verwalten – Datenbanksysteme 7.2 Daten anordnen mit Tabellen Projekteinstieg: Tabellen 7.3 Daten filtern mit SQL 7.4 Komplexe Filter
2. Modellierung von relationalen Datenbanken a) Datenbankentwurf durch ER-Diagramme - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung einer Datenbankmodellierung b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf - Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln c) Normalformen - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)		7.5 Datenbankentwurf 7.6 Umsetzung des ER-Modells Wiederaufgriff des Projekteinstiegs 7.7 Datenbanken verbessern durch Normalformen
3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken		7.8 Prüfungsvorbereitung

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

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

3.1. Zusammenarbeit mit anderen Fächern

Durch die unterschiedliche Belegung von Fächern können Schüler Aspekte aus anderen Kursen mit in den Informatikunterricht einfließen lassen. Es wird Wert darauf gelegt, dass in bestimmten Fragestellungen die Expertise einzelner Schüler gesucht wird, die aus einem von ihnen belegten Fach genauere Kenntnisse mitbringen und den Unterricht dadurch bereichern.

3.2. Vorbereitung auf die Erstellung der Facharbeit

Um eine einheitliche Grundlage für die Erstellung und Bewertung der Facharbeiten in der Jahrgangsstufe Q1 zu gewährleisten, findet im Vorfeld des Bearbeitungszeitraums ein fächerübergreifender Projektunterricht statt, in dem die schulinternen Richtlinien für Erstellung einer Facharbeit erläutert und die unterschiedlichen Arbeitsweisen in den wissenschaftlichen Fachbereichen berücksichtigt werden. Außerdem findet eine Schulung zur Umsetzung der Vorgaben der Formate der Facharbeit für alle Schüler der Jahrgangsstufe Q1 statt (Formatvorlagen). Ein dritter Baustein umfasst die Vorstellung und Diskussion lernbiologischer Grundlagen.

3.3. Exkursionen / Workshops

In der gymnasialen Oberstufe sollen in Absprache mit der Stufenleitung nach Möglichkeit unterrichtsbegleitende Exkursionen durchgeführt werden. Diese werden im Unterricht vor- bzw. nachbereitet. Die Fachkonferenz hält folgende Exkursionen für sinnvoll:

- RWTH Aachen, Infosphere: Alice, Informatik enlightened, Künstliche Intelligenz, Hausautomation, Media Computing, Medienmanipulation, Zelluläre Automaten, Newton meets Java...
- Besuch des Heinz Nixdorf Forum Museums in Paderborn

3.4. Evaluation des schulinternen Curriculums

Das schulinterne Curriculum stellt keine starre Größe dar, sondern ist als „lebendes Dokument“ zu betrachten. Dementsprechend werden die Inhalte stetig überprüft, um ggf. Modifikationen vornehmen zu können. Die Fachkonferenz trägt durch diesen Prozess zur Qualitätsentwicklung und damit zur Qualitätssicherung des Faches Informatik bei.

Die Evaluation erfolgt jährlich. Zu Schuljahresbeginn werden die Erfahrungen des vergangenen Schuljahres in der Fachschaft gesammelt, bewertet und eventuell notwendige Konsequenzen und Handlungsschwerpunkte formuliert.

3.5. Fachgruppenarbeit

Eine Checkliste im Fachbereich Informatik der Schul-Cloud dient dazu, den Ist-Zustand bzw. auch Handlungsbedarf in der fachlichen Arbeit festzustellen und zu dokumentieren, Beschlüsse der Fachkonferenz zur Fachgruppenarbeit in übersichtlicher Form festzuhalten sowie die Durchführung der Beschlüsse zu kontrollieren und zu reflektieren. Die Liste wird regelmäßig überarbeitet und angepasst. Sie dient auch dazu, Handlungsschwerpunkte für die Fachgruppe zu identifizieren und abzusprechen.

4 Grundsätze der Leistungsbewertung und -rückmeldung

4.1. Rechtliche Grundlagen und Prinzipien

Transparenz der Leistungsbeurteilung

Schulische Leistungsbewertung steht im Spannungsfeld pädagogischer und gesellschaftlicher Zielsetzung.

Unter pädagogischen Gesichtspunkten hat sie vornehmlich das Individuum im Blick. Hier soll sie über den Leistungszuwachs rückmelden und dadurch die Motivation für weitere Anstrengungen erhöhen. Sie ermöglicht den Schülern ihre noch vorhandenen fachlichen Defizite wie auch ihre Stärken und Fähigkeiten zu erkennen um dadurch ein realistisches Selbstbild aufzubauen. Sie ist Basis für gezielte individuelle Förderung.

Für die Erziehungsberechtigten und Schüler sind Noten eine einfache und zentrale Information über den Leistungsstand. Sie bieten Anlass, über die Ursache von Defiziten und über die Beseitigung von Lernschwierigkeiten verschiedenster Art Rücksprache zu halten. Noten sind zudem Grundlage und Anlass, in den halbjährlich stattfindenden Konferenzen über die Schwierigkeiten und besonderen Probleme einzelner Schüler wie auch Klassen zu beraten und Maßnahmen zur Verbesserung zu beschließen.

Schulische Leistungsbewertung ist eingebettet in die durch das Schulgesetz § 48 (Grundsätze der Leistungsbewertung), APO - GOST §13 bis §17 sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe vorgegebenen Grundsätze und Verfahren. Daraus erwächst für die Schulen konkret die Aufgabe, sowohl die individuellen Schwächen und Stärken der Schüler zu diagnostizieren und gegebenenfalls die Defizite durch gezielte Maßnahmen zu beseitigen sowie besondere Begabungen zu fördern.

Zudem haben Noten eine wichtige gesellschaftliche Funktion. Diese mit Inhalt zu füllen, ist Aufgabe der Schulen. Noten dienen als zentrale Entscheidungsgrundlage über Schullaufbahnen, Versetzungen und Abschlüsse. Zeugnisse sind ein entscheidender Parameter bei der Zuteilung von Berufs- und Lebenschancen. Daraus erwachsen für die Beurteilenden eine besondere Verantwortung und die Pflicht einer größtmöglichen Objektivität bei der Notenfindung.

Die Fachkonferenz Informatik legt die Kriterien für die Leistungsbeurteilung fest. Die Lehrer machen diese Kriterien den Schülern transparent.

Grundsätze der Leistungsbewertung

Es gelten folgende Grundsätze der Leistungsbewertung:

- Lernerfolgsüberprüfungen sind ein kontinuierlicher Prozess. Bewertet werden alle im Zusammenhang mit dem Unterricht erbrachten Leistungen (schriftliche Arbeiten, mündliche Beiträge, praktische Leistungen).
- Leistungsbewertung bezieht sich auf die im Unterricht erworbenen und geförderten Kompetenzen.
- Die Lehrperson gibt den Schülern im Unterricht hinreichend Gelegenheit, die entsprechenden Anforderungen der Leistungsbewertung im Unterricht in Umfang und Anspruch kennenzulernen und sich auf sie vorzubereiten.
- Bewertet werden der Umfang, die selbstständige und richtige Anwendung der Kenntnisse, Fähigkeiten und Fertigkeiten sowie die Art der Darstellung.

4.2. Anforderungen und Kriterien zur Beurteilung (EF – Q2)

Klausuren

Ab der Stufe EF werden gemäß der Vorgaben der Lehrpläne, der Vorgaben für das Abitur und des schulinternen Curriculums Klausuren geschrieben.

Anzahl und zeitlicher Umfang der Klausuren

Halbjahr	Grundkurs		Leistungskurs		Hinweise
	Anzahl	Dauer	Anzahl	Dauer	
EF/I	1	2	-	-	
EF/II	1	2	-	-	
Q1/I	2	2	2	3	
Q1/II	2	2	2	3	Die 1. Klausur kann durch eine Facharbeit ersetzt werden.
Q2/I	2	3	2	4	
Q2/II	1	3	1	4,15	Im GK nur für Schüler, die Informatik als 3. Abiturfach gewählt haben.

Mit Ausnahme der Klausur in Q2/II handelt es sich um Unterrichtsstunden, in Q2/II um Zeitstunden. Die Facharbeit in Q1 ersetzt die 1. Klausur in diesem Halbjahr. Die Verpflichtung, Klausuren in Fächern nach Wahl zu schreiben, gilt mindestens für ein Halbjahr.

Den Klausuren wird folgende Verteilung der Anforderungsbereiche zugrunde gelegt:

- Anforderungsbereich I: ca. 30 % (Reproduktion – Wiedergabe von Kenntnissen und Modellen, Beschreibung unter Verwendung gelernter Arbeitstechniken (Grafen, Tabellen, etc.))
- Anforderungsbereich II: ca. 50 % (Transfer – selbständiges Auswählen, Anordnen, Verarbeiten und Darstellen bekannter Sachverhalte unter vorgegebenen, neuen Gesichtspunkten, die aber in einem eingeübten Zusammenhang stehen; Auswerten von unbekanntem Material unter bekanntem Aspekt)
- Anforderungsbereich III: ca. 20 % (problemlösendes Denken – planmäßiges Verarbeiten komplexer Gegebenheiten mit dem Ziel selbständiger Lösung, Gestaltung, Deutung, Beurteilung;

dabei werden gelernte Methoden zur Lösung der Aufgabe selbständig neu kombiniert bzw. verändert, um sie der neuen Problemstellung anzupassen)

In den Aufgabenstellungen der Klausuren sollen die im Zentralabitur verwendeten Operatoren ab der Stufe EF verwendet werden, damit diese den Schülern von Beginn an vertraut sind.

Grundsätzlich richtet sich die Korrektur nach den Vorlagen, die aus den bisher durchgeführten Klausuren des Zentralabiturs bekannt sind. Sie muss für die Schüler nachvollziehbar sein. Wenn formale Korrekturzeichen nicht genügen, dann sind sie durch sachbezogene Hinweise am Rand oder am Ende der Arbeit zu ergänzen. Die Benotung der Klausuren in der S II soll sich an folgendem Punktesystem orientieren:

Punktesystem zur Benotung von Klausuren

Punkte in [%]	Punkte (Note)	Note
0	0	6
20	1	5 minus
26	2	5
33	3	5 plus
40	4	4 minus
45	5	4
50	6	4 plus
55	7	3 minus
60	8	3
65	9	3 plus
70	10	2 minus
75	11	2
80	12	2 plus
85	13	1 minus
90	14	1
95	15	1 plus

Die Begründung einer Note beinhaltet eine knappe Darstellung der positiven und negativen Anteile der Arbeit in den einzelnen Anforderungsbereichen, eine Information über Lernerfolg, –defizite und die Verwendung von Fachsprache.

Um die Leistungsbewertung durch die Fachkollegen und die Anforderungen an die Schüler zu vereinheitlichen, werden die Klausuraufgaben im Fachkollegium ausgetauscht und hinsichtlich des Anforderungsprofils überprüft. Darüber hinaus findet ein regelmäßiger Austausch von parallel unterrichtenden Kollegen statt. Hierbei werden methodische Schwerpunkte und grundlegende Bewertungskriterien vereinbart, die ein einheitliches Anforderungsprofil sicherstellen.

Sonstige Mitarbeit

Folgendes Bewertungsschema auf der Grundlage der Anforderungen des Lehrplans kann als Orientierung für die Bewertung der Sonstigen Mitarbeit dienen und bei den Schülern Transparenz schaffen.



Kriterium	Stufe 1	Stufe 2	Stufe 3	Stufe 4
Qualität der Beiträge	Meine Beiträge sind oft fachlich falsch, ich begründe meine Aussagen nicht	... manchmal fachlich falsch, ich begründe nur ansatzweise	... meist fachlich richtig und ich kann meine Aussagen begründen	... (fast) immer fachlich richtig, ich begründe und kann argumentieren
Beteiligung am Unterrichtsgespräch	Ich nehme nie unaufgefordert am Unterrichtsgespräch teil	... selten am Unterrichtsgespräch teil	... regelmäßig am Unterrichtsgespräch teil	... regelmäßig am Unterrichtsgespräch teil, ich habe gute Ideen
Aufmerksamkeit	Ich bin oft unaufmerksam	...gelegentlich unaufmerksam	... zumeist aufmerksam	... immer aufmerksam
Praktisches Arbeiten	Ich lese und befolge die Anleitungen nicht genau, hantiere unsachgemäß mit dem Material und komme in der Regel nicht zu den erwarteten Ergebnissen	... meist nicht so genau, gehe nicht nur zielgerichtet mit dem Material um und komme öfter nicht zu den erwarteten Ergebnissen	... meist genau, kann mit dem Material in der Regel sachgerecht umgehen und komme meist zu den erwarteten Ergebnissen	... genau, kann mit dem Material sachgerecht umgehen und komme zu guten Ergebnissen
Selbstständigkeit	Ich hole Rückstände nicht selbstständig auf, frage nie nach	Ich frage selten nach oder frage unnötiges nach	Ich frage nach, wenn es notwendig ist	Ich weiß, was zu tun ist und tue es auch, frage nach, wenn es notwendig ist
Gruppenarbeit	Ich halte andere eher von der Arbeit ab, rede über anderes, lenke ab; ich übernehme keine Präsentation des Ergebnisses	Ich arbeite nicht so richtig mit, störe andere aber nicht bei der Arbeit, ich präsentiere nur sehr ungern die Arbeitsergebnisse	Ich arbeite kooperativ in der Gruppe und schließe mich bereitwillig den anderen an; wenn gewünscht, präsentiere ich auch die Ergebnisse	Ich arbeite kooperativ in der Gruppe, mache Vorschläge für die Arbeit, Sorge für ein angenehmes Arbeitsklima und präsentiere gerne die Ergebnisse
Eigeninitiative	Ich habe Probleme, mit der Arbeit zu beginnen und konzentriert zu arbeiten	Ich arbeite erst auf Aufforderung und nicht immer konzentriert	Ich beginne zügig mit der Arbeit und arbeite die meiste Zeit konzentriert	Ich beginne direkt und bleibe konzentriert bei der Arbeit
Arbeitsorganisation	Arbeitsmaterial ist oft nicht vollständig oder ungeordnet	... normalerweise vorhanden, aber nicht sofort nutzbar	... normalerweise vorhanden und schnell nutzbar	... immer vorhanden, und sofort nutzbar
Hausaufgaben	Unterrichtsbeiträge auf Basis der Hausaufgaben kann ich nicht liefern, da ich die Hausaufgaben häufig nicht mache	... nur manchmal mache und wenn ja, da sie unvollständig sind.	... fast immer vollständig mache, sie aber nicht immer richtig sind	... immer vollständig mache und mir nur selten Fehler unterlaufen
Pünktlichkeit	Ich komme häufig zu spät	... pünktlich	... pünktlich und habe die Unterlagen auf dem Tisch	... pünktlich und bin sofort arbeitsbereit

Bei den mündlichen Leistungen im Unterricht fließen neben der Beteiligung am Unterrichtsgespräch auch Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts, die Präsentation von Arbeitsergebnissen mit ein, ebenso die Richtigkeit, Vollständigkeit und Komplexität der

Gedankengänge, die der Altersstufe angemessene sprachliche Darstellung und die Verwendung der Fachsprache. Bei der Unterrichtsgestaltung bieten sich den Schülern hinreichend Möglichkeiten zur Mitarbeit zu eröffnen, z.B. durch praktische Leistungen am Computer als Werkzeug im Unterricht, Protokolle und Referate, Projektarbeit, Abgabe von Lernprodukten in der Schul-Cloud sowie Lernerfolgsüberprüfungen und schriftliche Übungen.

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 Reflektion zu legen.

Bei Gruppenarbeiten wird 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 geachtet.

Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Jahres den Schülern transparent gemacht. In der Sekundarstufe II werden die Schüler etwa in der Mitte eines Halbjahres über ihren Leistungsstand informiert. 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- und Schülersprechtagen.

Die Leistungsrückmeldung kann erfolgen

-
- durch ein Gespräch mit der Schülerin oder dem Schüler,
 - durch einen Feedbackbogen,
 - durch die Schul-Cloud,
 - durch die schriftliche Begründung einer Note oder
 - durch eine individuelle Lern- und Förderempfehlung.

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

4.3. Individuelle Förderung

Die Lehrer beobachten die individuellen Leistungen in allen Bereichen der Informatik über einen längeren Zeitraum, um auf dieser Grundlage ein Leistungsbild zu erhalten. Neben der Orientierung an den Kompetenzstandards der jeweiligen Jahrgangsstufe kann bei der Leistungsbewertung auch die jeweilige Entwicklung des Schülers, gemäß der zu beobachtenden Lern- und Denkfortschritte, berücksichtigt werden.

Der Informatikunterricht lebt von der verantwortungsvollen und selbständigen Arbeit der Schüler, so dass die Lehrperson die nötige Zeit hat, bei Bedarf gezielt und individuell zu fördern.

Leistungsstärkere Schüler können ihr Wissen anhand von vertiefenden Problemstellungen erweitern oder als Experten für einzelne (Teil-) Probleme im Rahmen des Lernens durch Lehren ihren Mitschülern beratend zur Seite stehen.

4.4. Bildung der Zeugnisnote

In die Note gehen alle im Unterricht erbrachten Leistungen ein. Dabei nehmen die Beurteilung der Kursarbeiten bzw. Klausuren den gleichen Stellenwert wie die Leistungen im Bereich der Mitarbeit im Unterricht ein. Zudem ist bei der Notenfindung die individuelle Lernentwicklung der Schüler angemessen zu berücksichtigen.