

Fachvereinbarungen Informatik

Stand: August 2024



By Arhitectdanielheller - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=57531506>

Inhaltsverzeichnis

Grundsätze	4
Rahmenbedingungen der fachlichen Arbeit	4
Lehr- und Lernmittel.....	5
Leistungsbewertung	6
Schriftliche und praktische Leistungen	7
Mündliche und sonstige Leistungen.....	7
Ermittlung der Gesamtnote zum Halbjahr bzw. zum Schuljahresende	8
Korrekturzeichen	8
Hausaufgaben.....	9
Förderung	11
Spezielle Regelungen für die Sekundarstufe I	11
Leistungsbewertung	11
Schulinterner Lehrplan	11
5. Klasse: ENW	11
6. Klasse: Informatik	12
7. Klasse: Arbeitsgemeinschaft.....	20
9. + 10. Klasse: Wahlpflichtbereich II (Differenzierungskurs).....	20
Spezielle Regelungen für Sekundarstufe II	26
Leistungsbewertung	26
Schriftliche Arbeiten (Klausuren)	26
Sonstige Mitarbeit	27
Schulinterner Lehrplan	27
Entscheidungen zum Unterricht.....	27
Unterrichtsvorhaben	27
Didaktische Lernumgebung.....	28
Übersichtsraster Unterrichtsvorhaben EF.....	30
Konkretisierung der Unterrichtsvorhaben der EF	30
Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase	42
Konkretisierung der Unterrichtsvorhaben der Q1 (Grundkurs)	43
Konkretisierung der Unterrichtsvorhaben der Q2 (Grundkurs).....	54
Facharbeit.....	61
Außerunterrichtliches.....	61
Exkursionen	61
Wettbewerbe	61

Weitere Fachvereinbarungen.....	61
Sammlung.....	61
Fachschaftsordner	61

Grundsätze

Rahmenbedingungen der fachlichen Arbeit

Das Fach Informatik wird am OHG in den folgenden Jahrgangsstufen angeboten:

Klasse	Umfang (pro Woche)	
5	2 Stunden	ein Quartal im Rahmen des Faches ENW
6	2 Stunden	ganzjährig
7		AG
8		
9	3 Stunden	Differenzierungskurs (WP II)
10	3 Stunden	Differenzierungskurs (WP II)
EF	3 Stunden	Grundkurs
Q1	3 Stunden	Grundkurs
Q2	3 Stunden	Grundkurs

Der Unterricht erfolgt im 45-Minuten-Takt. Die Kursblockung sieht grundsätzlich für Grundkurse eine Doppelstunde und eine Einzelstunde vor.

In der Sekundarstufe II kann der Kurs mündlich oder schriftlich bzw. auch als 3. oder 4. Abiturfach gewählt werden.

Neben den normalen Unterrichtsangeboten gibt es für die Schülerinnen und Schüler die Möglichkeit an der „Homepage-AG“ teilzunehmen und Erfahrungen in der modernen Webentwicklung zu sammeln. Als MINT-EC Schule unterstützt die Informatik bei Bedarf die Projekte der Junior-Ingenieurs-Akademie.

Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in den Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind. Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei zusätzlich eine didaktische Bibliothek¹ zum Einsatz, welche das Erstellen von grafischen Programmen erleichtert.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern. Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Es wird grundsätzlich frei erhältliche Software bevorzugt, unter anderen, um Schülerinnen und Schüler eine Vor- und Nachbereitung des Unterrichts zu Hause zu erleichtern.

Für den Unterricht stehen zwei Computerräume mit 28 bis 32 PCs sowie jeweils fünf mobile Laptop- / iPad-Wagen zur Verfügung. Alle Schülerinnen und Schüler besitzen einen Office365 – Zugang, MS

¹ Es wird die im Lehrbuch Informatik 1 verwendete Greenfoot-Bibliothek benutzt.

Teams und OneNote werden im Informatikunterricht begleitend für Materialien und Aufgaben eingesetzt. Alle Computer sind an das schulinterne Netz angeschlossen, so dass die Schülerinnen und Schüler an allen Arbeitsplätzen und Geräten Zugriff auf ihre eigenen Daten haben und für Recherchen oder zur Bearbeitung schulischer Aufgaben verwenden können.

Lehr- und Lernmittel

Lehrbücher:

Klasse	Buch	ISBN
5		
6	Buchner – Verlag: Informatik 5/6	978 – 3 – 661 – 38041 – 4
7		
8		
9		
10		
EF	Schöningh-Verlag: Informatik 1	978 – 3 – 14 – 037126 – 1 978 – 3 – 14 – 123370 – 4 ²
Q1	Schöningh-Verlag: Informatik 2	978 – 3 – 14 – 037127 – 8
Q2	Schöningh-Verlag: Informatik 2	978 – 3 – 14 – 037127 – 8

Zu den verschiedenen Themen in der Differenzierung (Stufe 9 + 10) existiert selbst erstelltes und aktuell gehaltenes Unterrichtsmaterial. Die Unterrichtsmaterialien in allen anderen Stufen werden ebenfalls durch eigene digitale Unterrichtsmaterialien ergänzt, die dann jeweils digital zur Verfügung gestellt werden.

Als Präsentationssoftware wird überwiegend PowerPoint eingesetzt. Weitere Lernmittel in der EF sind verschiedene Software - Tools wie Greenfoot, BlueJ, der PAP-Designer, Violet oder Umllet als UML - Editor und ein Struktogrammeditor, um die Wichtigsten zu nennen. Natürlich gehört auch ein Computer mit einem internetfähigen Browser und einem Graphikprogramm (z.B. Paint oder Gimp) zum benötigten Unterrichtsmaterial.

Neben den in der EF zu benutzenden Tools gehören in der Q1 und Q2 als integrierte Programmierumgebung IntelliJ bzw. Visual Studio.

Im Themenbereich „Automaten und Grammatiken“ werden freie online verfügbare Tools wie <https://hepburn.codes/fsm-workbench/create.html>, <https://flaci.com/autoedit> und <https://flaci.com/home/> eingesetzt.

Als Programm für die Entwicklung einer Vernetzung wird FILIUS eingesetzt. Im Themenbereich Datenbanken werden Excel, Access und MySQL mit XAMPP-Server verwendet.

Für Projektarbeiten wird das Versionsverwaltungstool Git und die Internetplattform Trello genutzt.

² Neudruck 2021

Unterrichtsmaterial:

Fach	Klasse	Arbeitsheft	Klassenarbeitsheft	zusätzliches Material
ENW	5	nein	Projekte	Calliope https://code-it-studio.de/
Informatik	6	nein		Lehrbuch
AG	7	nein	Projekte	Office
	8			
Diff-kurse Informatik	9	nein	Nein	Software, Skripte
Diff-kurse Informatik	10	nein	Nein	Software, Skripte
Informatik	OS	Ordner	Nein	Software, Skripte, Lehrbuch

Leistungsbewertung

Die Leistungsbewertung richtet sich nach den Kompetenzbereichen und Inhaltsfeldern des Faches Informatik aus:

Kompetenzbereiche (Prozesse)	Inhaltsfelder (Gegenstände)
„Argumentieren“ „Modellieren“ „Implementieren“ „Darstellen und Interpretieren“ „Kommunizieren und Kooperieren“	„Daten und ihre Strukturierung“ „Algorithmen“ „Formale Sprachen und Automaten“ „Informatiksysteme“ „Informatik, Mensch und Gesellschaft“

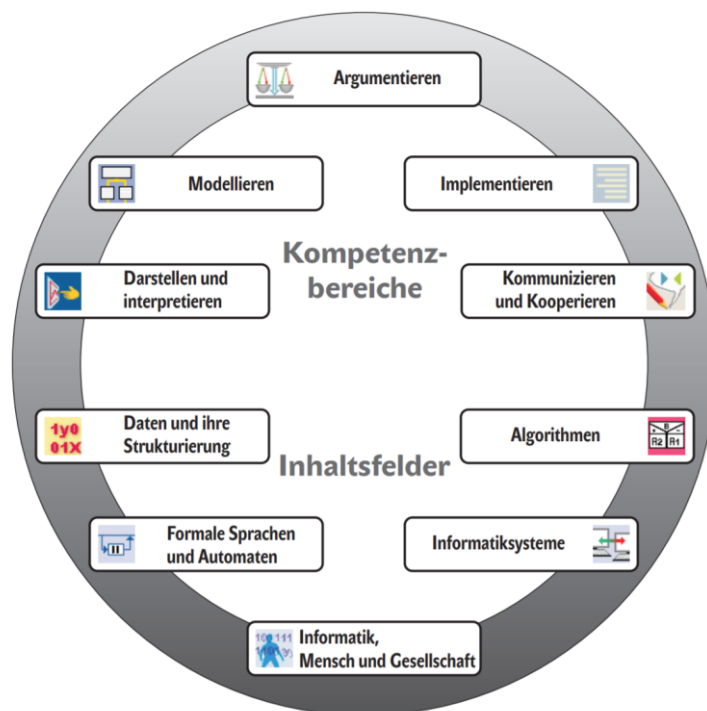


Abbildung 1: Übersicht Kompetenzen und Inhaltsfelder:

https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/KLP_GOST_Informatik.pdf (Seite 19)

Schriftliche und praktische Leistungen

Zu den praktischen Arbeiten gehört der komplette Durchlauf der Softwareentwicklung in dem alle oben genannten Kompetenzen in den jeweiligen Phasen der Erstellung eine Rolle spielen. So werden Problemlösungen zunächst modelliert und dann implementiert und getestet. Die Arbeit verläuft bei größeren Aufgaben als projektorientierte Gruppenarbeit ab, in der viel kommuniziert und kooperiert werden muss. Die erstellten Programme werden exemplarisch vorgestellt und die Güte der Lösung nach Gesichtspunkten wie Lesbarkeit und Funktionalität diskutiert. Schriftliche Leistungen finden in der Oberstufe und teilweise in der Sekundarstufe I in Form von schriftlichen Klassenarbeiten bzw. Klausuren statt.

Mündliche und sonstige Leistungen

Im Rahmen des Unterrichts werden schriftliche Arbeiten wie Präsentationen, Referate und Recherchen erstellt und präsentiert. Die Bewertung der Beiträge hängt von der Qualität, Quantität und Kontinuität im Rahmen des Unterrichtsverlaufs ab. Diese Leistungen sind unabhängig von der Note der schriftlichen Leistungen. Bei der Bildung der Quartals- und Abschlussnote wird jeweils die Gesamtentwicklung der Schülerin bzw. des Schülers berücksichtigt.

Leistungsaspekt	Anforderungen für eine	
	gute Leistung	ausreichende Leistung
	<i>Die Schülerin, der Schüler</i>	
Qualität der Unterrichtsbeiträge	nennt richtige Lösungen und begründet sie nachvollziehbar im Zusammenhang der Aufgabenstellung	nennt teilweise richtige Lösungen, in der Regel jedoch ohne nachvollziehbare Begründungen
	geht selbstständig auf andere Lösungen ein, findet Argumente und Begründungen für ihre/seine eigenen Beiträge	geht selten auf andere Lösungen ein, nennt Argumente, kann sie aber nicht begründen
	kann ihre/seine Ergebnisse auf unterschiedliche Art und mit unterschiedlichen Medien darstellen	kann ihre/seine Ergebnisse nur auf eine Art darstellen
Kontinuität/Quantität	beteiligt sich regelmäßig am Unterrichtsgespräch	nimmt eher selten am Unterrichtsgespräch teil
Selbstständigkeit	bringt sich von sich aus in den Unterricht ein	beteiligt sich gelegentlich eigenständig am Unterricht
	ist selbstständig ausdauernd bei der Sache und erledigt Aufgaben gründlich und zuverlässig	benötigt oft eine Aufforderung, um mit der Arbeit zu beginnen; arbeitet Rückstände nur teilweise auf
	strukturiert und erarbeitet neue Lerninhalte weitgehend selbstständig, stellt selbstständig Nachfragen	erarbeitet neue Lerninhalte mit umfangreicher Hilfestellung, fragt diese aber nur selten nach
	erarbeitet bereitgestellte Materialien selbstständig	erarbeitet bereitgestellte Materialien eher lückenhaft

Hausaufgaben	erledigt sorgfältig und vollständig die Hausaufgaben	erledigt die Hausaufgaben weitgehend vollständig, aber teilweise oberflächlich
	trägt Hausaufgaben mit nachvollziehbaren Erläuterungen vor	nennt die Ergebnisse, erläutert erst auf Nachfragen und oft unvollständig
Kooperation	bringt sich ergebnisorientiert in die Gruppen-/Partnerarbeit ein	bringt sich nur wenig in die Gruppen-/Partnerarbeit ein
	arbeitet kooperativ und respektiert die Beiträge Anderer	unterstützt die Gruppenarbeit nur wenig, stört aber nicht
Gebrauch der Fachsprache	wendet Fachbegriffe sachangemessen an und kann ihre Bedeutung erklären	versteht Fachbegriffe nicht immer, kann sie teilweise nicht sachangemessen anwenden
Werkzeuggebrauch	setzt Werkzeuge im Unterricht sicher bei der Bearbeitung von Aufgaben und zur Visualisierung von Ergebnissen ein	benötigt häufig Hilfe beim Einsatz von Werkzeugen zur Bearbeitung von Aufgaben
Präsentation/Referat	präsentiert vollständig, strukturiert und gut nachvollziehbar	präsentiert an mehreren Stellen eher oberflächlich, die Präsentation weist Verständnislücken auf
Schriftliche Übung/Tests	ca. 75% der erreichbaren Punkte	ca. 50% der erreichbaren Punkte

Ermittlung der Gesamtnote zum Halbjahr bzw. zum Schuljahresende

Die „Sonstigen Leistungen“ bzw. die „Sonstige Mitarbeit“ haben den gleichen Stellenwert wie die schriftlichen Arbeiten. Daher sind **beide Leistungsbereiche etwa zur Hälfte** bei der Endnote zu berücksichtigen.

Auch in der Sekundarstufe I sollten die „Sonstigen Leistungen“ also in etwa gleichgewichtig zu den Klassenarbeiten gewertet werden, d.h. sowohl die schriftlichen Arbeiten als auch die „Sonstigen Leistungen“ sollten mindestens mit 40% in die Endnote einfließen.

Korrekturzeichen

Allgemeine Korrekturzeichen zum sachgemäßen Sprachgebrauch

Zeichen	Beschreibung
R	Rechtschreibung
Z	Zeichensetzung
G*	Grammatik (wenn nicht weiter spezifiziert, auch Syntax)
W **	Wortschatz

- * Zur Spezifizierung von Grammatik- und Syntaxfehlern stehen zudem folgende Korrekturzeichen zur Verfügung:

Zeichen	Beschreibung
T	Tempus
M	Modus
N	Numerus
Sb	Satzbau
St	Wortstellung
Bz	Bezug

- ** Zur Spezifizierung von Wortschatzfehlern stehen zudem folgende Korrekturzeichen zur Verfügung:

Zeichen	Beschreibung
A	Ausdruck/unpassende Stilebene o.ä.
FS	Fachsprache (fehlend/falsch)

Inhaltliche Korrekturzeichen

Zeichen	Beschreibung
✓	richtig (Ausführung/Lösung/etc.)
f	falsch (Ausführung/Lösung/etc.)
(✓)	folgerichtig (richtige Lösung auf Grundlage einer fehlerhaften Annahme/Zwischenlösung)
≈	ungenau (Ausführung/Lösung/etc.)
[—]	Streichung (überflüssiges Wort/Passage)
⌈ bzw. #	Auslassung
Wdh	Wiederholung, wenn vermeidbar

Hausaufgaben

Hausaufgaben sind ein Instrument selbst organisierten Lernens und der Festigung der im Unterricht gewonnenen Erkenntnisse. Sie dienen dazu, Wissen zu vertiefen, Neues zu erproben, Kenntnisse zu sichern und Verantwortung zu übernehmen, z.B. durch selbstständige Zeiteinteilung oder Informationsbeschaffung. Hausaufgaben dienen ferner als Brücke zum Elternhaus, da sie einen Einblick in die Arbeit der Schule geben und den Wissensstand der Kinder erkennen lassen.

Beim Erteilen von Hausaufgaben sind folgende Aspekte zu beachten:

- Sie bereiten auf den Unterricht vor.
- Sie bereiten den Unterricht nach.
- Sie erwachsen aus dem Unterricht.
- Sie führen an eigenständiges Arbeiten heran.

<i>Aufgaben der Beteiligten</i>	
Schülerinnen und Schüler	Lehrerinnen und Lehrer
• verpflichtendes Eintragen der Aufgaben in einem Aufgabenheft bzw. Timer • genaue Beachtung der Aufgabenstellung • eigene Zeitplanung • unaufgefordertes Melden bei nicht erledigten Hausaufgaben zu Beginn der Stunde und eigenständiges Nachreichen in der Folgestunde	• klare, eindeutige Formulierungen – in Klassen 5 + 6 zusätzlich durch Tafelanschrieb • rechtzeitiges Erteilen der Aufgaben vor Stundenende • gezielte Kontrolle und Rückmeldung • Absprache der Kollegen untereinander • Berücksichtigung des Nachmittagsunterrichts (keine Aufgaben zum folgenden Tag im gleichen Fach) • Klassenbucheintrag und Elternrückmeldung bei dreimaliger Nichterledigung durch den Fachlehrer (ggf. Notiz an den Klassenlehrer) • Versuch, im Gespräch die Gründe für das Nichtanfertigen zu ermitteln

Die Schülerinnen und Schüler sollen befähigt werden, komplexe Informatiksysteme sicher und verantwortungsvoll zu verwenden und die Chancen und Risiken dieser Informatiksysteme zu beurteilen und sich aktiv an der Fortentwicklung zu beteiligen.

Die Hausaufgaben sollen die Schülerinnen zum informatischen Problemlösen befähigt werden und ihr Abstraktionsvermögen und ihr strukturelles und kreatives Denken durch entsprechende Aufgaben fördern, indem sie selbstständig eigene Lösungen erarbeiten und im Plenum vorstellen.

Vereinbarte Richtwerte für die Dauer der Hausaufgaben im Fach Informatik	
Stufe 5 + 6	bis zu 20 Minuten pro Unterrichtsstunde
Stufe 7 – 10	bis zu 30 Minuten pro Unterrichtsstunde
EF	bis zu 30 Minuten pro Unterrichtsstunde
Q1 / Q2	bis zu 30 Minuten pro Unterrichtsstunde (Grundkurs) bis zu 40 Minuten pro Unterrichtsstunde (Leistungskurs)

Die Kollegen sprechen sich dahingehend ab, dass die Richtwerte für die allgemeinen Hausaufgaben in den jeweiligen Stufen nicht überschritten werden.

Förderung

Leistungsschwachen Schülerinnen und Schülern wird Selbstlernmaterial zum Beispiel zur Erlernung der Programmiersprache Java empfohlen.

Leistungsstarke Schülerinnen und Schüler werden auf zusätzliche Wettbewerbe, MINT-Angebote, Softwareprojekte oder besondere Lernleistungen aufmerksam gemacht.

Spezielle Regelungen für die Sekundarstufe I

Leistungsbewertung

In ENW und der Stufe 6 erfolgt die Leistungsbewertung in allen drei oben genannten Bereichen, wobei der Schwerpunkt auf projektorientiertem Arbeiten liegt.

Im Bereich des Wahlpflichtfachs in der 9. und 10. Klasse wird mindestens eine Klassenarbeit im Halbjahr geschrieben. Eine zweite Klassenarbeit im Halbjahr kann durch ein Projekt ersetzt werden.

Schulinterner Lehrplan

Die folgende Übersicht über die Unterrichtsvorhaben stellt für die Lehrerinnen und Lehrer einen verbindlichen inhaltlichen Rahmen dar. Die Übersicht dient dazu, sich schnell einen Überblick über Themen, Fragestellungen oder zu erweiternde Kompetenzen zu erhalten. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierung, die nach Bedarf über- oder unterschritten werden kann. Somit steht zusätzlicher pädagogischer Spielraum für Vertiefungen wie besondere Interessen der Lerngruppe, aktuelle Themen oder besondere Anlässe zur Verfügung.

5. Klasse: ENW

Dauer: Ein Quartal

Erste Schritte in die Welt der Programmierung mit Blockprogrammierung und dem Calliope Mini. Für die Unterrichtsvorhaben zur Steuerung von Automaten und Robotern wird die offene Programmierumgebung Open Roberta und der Calliope verwendet.

Übersicht über die Unterrichtsvorhaben in Klasse 6:

<u>UV 6 – I: Informatiksysteme</u> • Was ist das Grundprinzip der elektronischen Datenverarbeitung? • Welche Bestandteile hat ein Computer? Was ist der Unterschied zwischen Hardware und Software? • Wie kann ich Dateien und Dokumente auf dem Computer speichern und in Ordnern verwalten?	<u>UV 6 – II: Daten und Codierung</u> • Was ist der Zusammenhang zwischen „Daten“ und „Informationen“? Wie stelle ich Informationen geeignet dar? • Was bedeutet „Codierung“ von Daten? Wie kann ich selbst Codierungen von Daten durchführen? • Welchen Code verwenden Informatiksysteme? In welchen Einheiten werden Datenmengen ausgedrückt?
<u>UV 6 – III: Kryptologie</u> • Wie halte ich Nachrichten vor anderen geheim? Welche Möglichkeiten zur Verschlüsselung gibt es? • Welche Verschlüsselungsverfahren sind besonders sicher?	<u>UV 6 – IV: Algorithmen</u> • Was ist ein Algorithmus? Wie kann ich Anweisungen so formulieren, dass ein Computer sie versteht? • Wie kann ich Anweisungen so kurz wie möglich halten und zu einem Ablauf zusammenfassen?
<u>UV 6 – V: Programmieren mit Scratch</u> • Wie finde ich mich in einer visuellen Programmierungsumgebung zurecht? • Wie kann ich Figuren mit Anweisungen und Sequenzen steuern? Wie finde ich Fehler in meinem Programm? • Wie kann ich Wiederholungen in meinen Programmcode sinnvoll kürzen?	<u>UV 6 – VI: Automatisierung und KI</u> • Wo und wie helfen mir Automatisierung und künstliche Intelligenz im Alltag? • Wie werden automatische Abläufe in der Informatik beschrieben? • Wie können Maschinen lernen, auch unbekannte Abläufe zu lösen?
<u>UV 6 – VII: IF, Mensch und Gesellschaft</u> • Was ist eigentlich das Internet? Welche Daten speichert es und welche Gefahren lauern darin? • Wie kann ich mich und meine Daten schützen? Welche Regeln und Gesetze gelten im Internet? • Welchen Einfluss nehmen Soziale Medien auf unser heutiges Leben?	

Konkretisierung der Unterrichtsvorhaben in Klasse 6:

UV 6 – I: Informatiksysteme

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Information und Daten - Informationsgehalt von Daten Informatiksysteme - Aufbau und Funktionsweise von Informatiksystemen - Anwendung von Informatiksystemen Informatik, Mensch und Gesellschaft - Informatiksysteme in der Lebens- und Arbeitswelt - Datenbewusstsein Themen: Das EVA-Prinzip Hardware und Software Benutzerkonten und Passwörter Speichern und Ordnen Ordnerstrukturen Arten der Datenspeicherung Umfang: ca. 8 Stunden	Argumentieren (A) - formulieren Fragen zu einfachen informatischen Sachverhalten - äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen - begründen die Auswahl eines Informatiksystems Modellieren und Implementieren (MI) - erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) - beschreiben einfache Darstellungen von informatischen Sachverhalten Kommunizieren und Kooperieren (KK) - erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht - setzen bei der Bearbeitung einer informatischen Problemstellung geeignete digitale Werkzeuge zum kollaborativen Arbeiten ein (MKR 1.2, 3.1)	- stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI) - interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI) - benennen Beispiele für (vernetzte) Informatiksysteme aus ihrer Erfahrungswelt (DI) - benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI) - beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung (DI) (MKR 6.1) - vergleichen Möglichkeiten der Datenverwaltung hinsichtlich ihrer spezifischen Charakteristika (u. a. Speicherort, Kapazität, Aspekte der Datensicherheit) (A) - setzen zielgerichtet Informatiksysteme zur Verarbeitung von Daten ein (MI) - erläutern Prinzipien der strukturierten Dateiverwaltung (A) (MKR 1.3) - setzen Informatiksysteme zur Kommunikation und Kooperation ein (KK) (MKR 3.1) - beschreiben an Beispielen die Bedeutung von Informatiksystemen in der Lebens- und Arbeitswelt (KK) (MKR 6.4) - erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK) - erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A) (VB C Z3)

Weitere Hinweise, Vereinbarungen und Absprachen:

- mögliches Projekt: einen Computer auseinander- und wieder zusammenbauen

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Information und Daten ▪ Daten und ihre Codierung ▪ Informationsgehalt von Daten Algorithmen ▪ Algorithmen und algorithmische Grundkonzepte Informatiksysteme ▪ Aufbau und Funktionsweise von Informatiksystemen Informatik, Mensch und Gesellschaft ▪ Datenbewusstsein Themen: Kommunikation im Alltag und in der Informatik Arten der Codierung Bits und Bytes Binärzahlen ASCII-Code Umfang: ca. 6 Stunden	Argumentieren (A) ▪ formulieren Fragen zu einfachen informatischen Sachverhalten ▪ äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen Modellieren und Implementieren (MI) ▪ erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) ▪ beschreiben einfache Darstellungen von informatischen Sachverhalten ▪ stellen informatische Sachverhalte in geeigneter Form dar ▪ interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) ▪ erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht ▪ kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme	▪ erläutern den Datenbegriff anhand von Beispielen aus ihrer Erfahrungswelt (A) ▪ erläutern den Zusammenhang und die Bedeutung von Information und Daten (A) ▪ stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder grafisch dar (DI) ▪ nennen Beispiele für die Codierung von Daten aus ihrer Erfahrungswelt (DI) ▪ codieren und decodieren Daten unter Verwendung des Binärsystems (MI) ▪ interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI) ▪ erläutern Einheiten von Datenmengen (A / KK) ▪ vergleichen Datenmengen hinsichtlich ihrer Größe mit Hilfe anschaulicher Beispiele aus ihrer Lebenswelt (DI)

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Information und Daten ▪ Daten und ihre ▪ Codierung ▪ Verschlüsselungs ▪ verfahren Algorithmen ▪ Algorithmen und algorithmische Grundkonzepte Informatiksysteme ▪ Aufbau und Funktionsweise von Informatiksystemen Informatik, Mensch und Gesellschaft ▪ Datenbewusstsein ▪ Datensicherheit und Sicherheitsregeln Themen: Verschlüsselung Substitution und Transposition Überschlüsselung Umfang: ca. 6 Stunden	Argumentieren (A) ▪ formulieren Fragen zu einfachen informatischen Sachverhalten ▪ äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen Modellieren und Implementieren (MI) ▪ erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) ▪ beschreiben einfache Darstellungen von informatischen Sachverhalten ▪ stellen informatische Sachverhalte in geeigneter Form dar ▪ interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) ▪ erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht ▪ kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme ▪ strukturieren gemeinsam eine Lösung für ein informatisches Problem	▪ erläutern ein einfaches Transpositionsverfahren als Möglichkeit der Verschlüsselung (DI) (MKR 1.4) ▪ vergleichen verschiedene Verschlüsselungsverfahren unter Berücksichtigung von ausgewählten Sicherheitsaspekten (DI) (MKR 1.4) ▪ führen Handlungsvorschriften schrittweise aus (Mi) ▪ beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A)

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Information und Daten ▪ Daten und ihre Codierung ▪ Informationsgehalt von Daten Algorithmen ▪ Algorithmen und algorithmische Grundkonzepte ▪ Implementation von Algorithmen Themen: Beschreibung von Abläufen Algorithmen im Alltag Genaue Anweisungen und Abläufe Bedingte Anweisungen und Verzweigungen Vom Algorithmus zum Programm Umfang: ca. 12 Stunden	Argumentieren (A) ▪ formulieren Fragen zu einfachen informatischen Sachverhalten ▪ äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen ▪ erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) ▪ beschreiben einfache Darstellungen von informatischen Sachverhalten Kommunizieren und Kooperieren (KK) ▪ erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht ▪ setzen bei der Bearbeitung einer informatischen Problemstellung geeignete digitale Werkzeuge zum kollaborativen Arbeiten ein (MKR 1.2, 3.1)	▪ stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI) ▪ formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften (DI) ▪ überführen Handlungsvorschriften in einen Programmablaufplan (PAP) oder ein Struktogramm (MI) ▪ führen Handlungsvorschriften schrittweise aus (MI) ▪ identifizieren in Handlungsvorschriften Anweisungen und die algorithmischen Grundstrukturen Sequenz, Verzweigung und Schleife (MI) (MKR 6.2) ▪ implementieren Algorithmen in einer visuellen Programmiersprache (MI) (MKR 6.1, 6.3) ▪ implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung (MI), (MKR 6.1, 6.3) ▪ überprüfen die Wirkungsweise eines Algorithmus durch zielgerichtetes Testen (MI) (MKR 6.2)

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Information und Daten ▪ Informationsgehalt von Daten Algorithmen ▪ Algorithmen und algorithmische Grundkonzepte ▪ Implementation von Algorithmen Informatiksysteme ▪ Aufbau und Funktionsweise von Informatiksystemen Themen: Einführung in die Programmierung mit Scratch Anweisungen und Sequenzen Wiederholungen und Verzweigungen Variablen Wiederholungen mit Abbruchbedingungen Umfang: ca. 12 Stunden	Argumentieren (A) ▪ bewerten ein Ergebnis einer informatischen Modellierung (MKR 6.4) Modellieren und Implementieren (MI) ▪ erstellen informatische Modelle zu gegebenen Sachverhalten ▪ implementieren informatische Modelle unter Verwendung algorithmischer Grundstrukturen (MKR 6.1, 6.2) ▪ überprüfen Modelle und Implementierungen Darstellen und Interpretieren (DI) ▪ beschreiben einfache Darstellungen von informatischen Sachverhalten ▪ stellen informatische Sachverhalte in geeigneter Form dar Kommunizieren und Kooperieren (KK) ▪ erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht ▪ kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme ▪ strukturieren gemeinsam eine Lösung für ein informatisches Problem	▪ stellen eine ausgewählte Information in geeigneter Form als Daten formalsprachlich oder graphisch dar (DI) ▪ interpretieren ausgewählte Daten als Information im gegebenen Kontext (DI) ▪ ermitteln durch die Analyse eines Algorithmus dessen Ergebnis (DI) (MKR 6.2) ▪ bewerten einen als Quelltext, Programmablaufplan (PAP) oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität (A) (MKR 6.3) ▪ benennen Grundkomponenten von Informatiksystem und beschreiben ihre Funktionen (DI) ▪ beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung (DI)

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Automaten und künstliche Intelligenz: ▪ Aufbau und Wirkungsweise einfacher Automaten ▪ Maschinelles Lernen mit Entscheidungsbäumen ▪ Maschinelles Lernen mit neuronalen Netzen Informatiksysteme ▪ Aufbau und Funktionsweise von Informatiksystemen Informatik, Mensch und Gesellschaft ▪ Informatiksysteme in der Lebens- und Arbeitswelt ▪ Datenbewusstsein Themen: Automaten im Alltag Zustandsdiagramme Künstliche Intelligenz in unserem Alltag Wiederholungen und Verzweigungen Entscheidungsbäume Neuronale Netze Umfang: ca. 10 Stunden	Argumentieren (A) ▪ formulieren Fragen zu einfachen informatischen Sachverhalten ▪ äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen ▪ erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen Modellieren und Implementieren (MI) ▪ erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) ▪ beschreiben einfache Darstellungen von informatischen Sachverhalten ▪ stellen informatische Sachverhalte in geeigneter Form dar ▪ interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) ▪ erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht	▪ erläutern die Funktionsweise eines Automaten aus ihrer Lebenswelt (A) (MKR 6.1) ▪ benennen Anwendungsbeispiele künstlicher Intelligenz aus ihrer Lebenswelt (A) ▪ stellen Abläufe in Automaten graphisch dar (DI) ▪ stellen das Grundprinzip eines Entscheidungsbaumes enaktiv als ein Prinzip des maschinellen Lernens dar (DI) ▪ beschreiben die grundlegende Funktionsweise künstlicher neuronaler Netze in verschiedenen Anwendungsbeispielen (KK) ▪ benennen Grundkomponenten von (vernetzten) Informatiksystemen und beschreiben ihre Funktionen (DI) ▪ erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK), ▪ beschreiben das Prinzip der Eingabe, Verarbeitung und Ausgabe (EVA-Prinzip) als grundlegendes Prinzip der Datenverarbeitung (DI)“

Inhaltliche Schwerpunkte	Übergeordnete Kompetenzerwartungen <i>Die Schüler*innen...</i>	Konkretisierte Kompetenzerwartungen <i>Die Schüler*innen...</i>
Inhaltsfelder: Informatik, Mensch und Gesellschaft ▪ Informatiksysteme in der Lebens- und Arbeitswelt ▪ Datenbewusstsein ▪ Datensicherheit und Sicherheitsregeln Themen: Kleine und große Netzwerke Daten und Gefahren im Internet Schutz von Daten Rechte von Nutzern Soziale Medien Umfang: ca. 6 Stunden	Argumentieren (A) ▪ formulieren Fragen zu einfachen informatischen Sachverhalten ▪ äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen ▪ erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen Modellieren und Implementieren (MI) ▪ erstellen informatische Modelle zu gegebenen Sachverhalten Darstellen und Interpretieren (DI) ▪ beschreiben einfache Darstellungen von informatischen Sachverhalten ▪ stellen informatische Sachverhalte in geeigneter Form dar ▪ interpretieren informatische Darstellungen Kommunizieren und Kooperieren (KK) ▪ erläutern informatische Sachverhalte unter Verwendung von Fachbegriffen sachgerecht ▪ kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme	▪ benennen an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen auf ihre Lebens- und Erfahrungswelt (A/KK) (MKR 6.4) / (VB C Z5) ▪ anstelle der vorherigen KE: erläutern an ausgewählten Beispielen Auswirkungen des Einsatzes von Informatiksystemen (A/KK), (MKR 6.4) / (VB C Z5) ▪ beschreiben anhand von ausgewählten Beispielen die Verarbeitung und Nutzung personenbezogener Daten (DI) (VB C Z5) ▪ erläutern anhand von Beispielen aus ihrer Lebenswelt Nutzen und Risiken beim Umgang mit eigenen und fremden Daten auch im Hinblick auf Speicherorte (A) (VB C Z3) ▪ beschreiben Maßnahmen zum Schutz von Daten mithilfe von Informatiksystemen (A), (MKR 1.4) / (VB C Z2)

7. Klasse: Arbeitsgemeinschaft

In der AG werden ihr Grafiken, Fotos, Musik und Ton in interaktiven Kreationen kreativ kombiniert. Mit **Scratch** können Charaktere geschaffen werden, die tanzen, singen und miteinander interagieren. Es wird ein tieferer Einblick in das Programmieren von Datenstrukturen wie Texten und Listen geschaffen sowie die vorhandenen Programmierkenntnisse anhand von kleinen Übungssequenzen und eigenen Projekten (Spiele, Simulationen und Animationen wie einen interaktiven Bericht oder einen Escape Room oder ein Quiz, Strategie- oder Geschicklichkeitsspiel) geübt und erweitert.

Zusätzlich zur Programmierung werden die Themen Bildbearbeitung und Urheberrecht behandelt, um ansprechende Graphiken für die verschiedenen Projekte zu erstellen.

Scratch ist eine grafische und frei erhältliche Programmierumgebung für Schülerinnen und Schüler (Informationen unter <http://scratch.mit.edu/>). Neben Scratch kommt die graphische Programmierumgebung des **Calliope Mini** zum Einsatz.

Als Nebeneffekt trainieren wir so für Informatikwettkämpfe wie den Jugendwettbewerb Informatik.

9. + 10. Klasse: Wahlpflichtbereich II (Differenzierungskurs)

Übersicht über die Unterrichtsvorhaben in Klassen 9 + 10:

UV 9 – I: Wie funktioniert unser Schulnetzwerk?	UV 9 – II: Das weltweite Datennetz – ein Geheimnis? Wir analysieren Webseiten und erstellen eigene Präsentationen für das Internet.
UV 9 – III: Mein digitaler Fußabdruck – wo hinterlasse ich Daten und was kann daraus geschlossen werden?	UV 9 – IV: Geheim ist geheim? Sichere Kommunikation mit Kryptographie
UV 9 – V: Helfer in Alltag und Arbeitswelt – wie werden Computer mit Hilfe von Sensoren und Aktoren selbständig? Wo spielen Computer in Alltagsgeräten eine Rolle?	
UV 10 – I: Computerprogramme mit System entwickeln – Einstieg in die textorientierte Programmierung	UV 10 – II: Der Blick in die Glaskugel - Simulation und Prognose mit Hilfe textorientierter Programmierung oder einer Tabellenkalkulation
UV 10 – III: Innenansichten des Computers - von der Software zur Hardware	UV 10 – IV: Das Internet der Dinge - Allgegenwärtige Informationstechnologien
UV 10 – V: Vertiefendes Projekt	

Konkretisierung der Unterrichtsvorhaben in den Klassen 9 + 10:

UV 9 – I: Wie funktioniert unser Schulnetzwerk?

(Zeit: ca. 3 UStd)

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Aufbau und Funktionsweise einfacher Informatiksysteme • Anwendung von Informatiksystemen • Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen

Dieses UV erweitert die in der Erprobungsstufe erworbenen Kompetenzen.

UV 9 – II: Das weltweite Datennetz – ein Geheimnis? Wir analysieren Webseiten und erstellen eigene Präsentationen für das Internet.

(Zeit: ca. 33 UStd)

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren	• Information und Daten • Formale Sprachen • Informatiksysteme • Informatik, Mensch und Gesellschaft	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Erstellung von Quelltexten • Analyse von Quelltexten • Aufbau und Funktionsweise einfacher Informatiksysteme • Anwendung von Informatiksystemen • Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen • Chancen und Risiken der Nutzung von Informatiksystemen

Erstellung von Internetseiten in HTML, Formatierung mithilfe von CSS, Erläuterung von rechtlichen Rahmenbedingungen für Veröffentlichungen. Erstellung einer Präsentation in Zusammenarbeit mit einem anderen Fach.

UV 9 – III: Mein digitaler Fußabdruck – wo hinterlasse ich Daten und was kann daraus geschlossen werden? **(Zeit: ca. 9 UStd)**

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Anwendung von Informatiksystemen • Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen • Chancen und Risiken bei der Nutzung von Informatiksystemen

Quellen für personenbezogene Informationen ermitteln, Verknüpfung personenbezogener Informationen aus verschiedenen Quellen, Chancen und Risiken verknüpfter Datenbestände, ausgewählte rechtliche Aspekte

UV 9 – IV: Geheim ist geheim? Sichere Kommunikation mit Kryptographie **(Zeit: ca. 9 UStd)**

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren	• Information und Daten • Algorithmen • Informatiksysteme • Informatik, Mensch und Gesellschaft	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Algorithmen entwerfen, darstellen und realisieren • Algorithmen analysieren und beurteilen • Anwendung von Informatiksystemen • Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen • Chancen und Risiken bei der Nutzung von Informatiksystemen

Klärung historischer Aspekte, Behandlung von Skytale und Cäsar-Verschlüsselung, aktuelle Möglichkeiten zum Schutz der eigenen Privatsphäre.

UV 9 – V: Helfer in Alltag und Arbeitswelt – wie werden Computer mit Hilfe von Sensoren und Aktoren selbständig? Wo spielen Computer in Alltagsgeräten eine Rolle?

(Zeit: ca. 36 UStd)

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Information und Daten • Informatiksysteme • Algorithmen • Informatik, Mensch und Gesellschaft	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Aufbau und Funktionsweise einfacher Informatiksysteme • Anwendung von Informatiksystemen • Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen • Chancen und Risiken bei der Nutzung von Informatiksystemen

UV 10 – I: Computerprogramme mit System entwickeln – Einstieg in die textorientierte Programmierung

(Zeit: ca. 30 UStd)

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren	• Information und Daten • Algorithmen • Formale Sprachen	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Entwurf von Algorithmen • Analyse von Algorithmen • Erstellung von Quelltexten • Analyse von Quelltexten

Die Fachkonferenz hat sich verbindlich auf die Programmiersprache Python geeinigt. Entwurf und Implementierung von Algorithmen, Verwendung von Kontrollstrukturen, Variablen, Methoden und Parametern, Verwendung eines strukturierten Datentyps, Strukturierung von Programmen, Analyse und Test von Programmen.

UV 10 – II: Der Blick in die Glaskugel - Simulation und Prognose mit Hilfe textorientierter Programmierung oder einer Tabellenkalkulation **(Zeit: ca. 9 UStd)**

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren	• Information und Daten • Künstliche Intelligenz und maschinelles Lernen • Informatik, Mensch und Gesellschaft	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Chancen und Risiken der Nutzung von Informatiksystemen • Beispiele künstlicher Intelligenz • Methoden maschinelles Lernens • Chancen und Risiken maschinellen Lernens

Visualisierung von Daten mit Diagrammen, Anwendung komplexer Formeln mit absoluter und relativer Adressierung, Tabellenkalkulation als Modellbildungs- und Simulationswerkzeug zum Vergleich unterschiedlicher Wachstumsmodelle, Chancen und Risiken von Simulationsmodellen.

Kooperation mit dem Mathematikunterricht bezüglich der theoretischen Grundlagen von Wachstumsmodellen.

UV 10 – III: Innenansichten des Computers - von der Software zur Hardware **(Zeit: ca. 24 UStd)**

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren und Implementieren • Darstellen und Interpretieren	• Information und Daten • Informatiksysteme	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Aufbau und Funktionsweise von Informatiksystemen und ihren Komponenten

EVA-Prinzip und Zuordnung der Hardware-Komponenten, Überblick über die Von-Neumann-Architektur, Zahldarstellungen und Grundrechenarten im Binärsystem, Simulation von logischen Schaltungen mit Logic Simulator.

Es werden Rechnermodelle im Sammlungsraum aufbewahrt, die die Lernenden zerlegen und analysieren.

UV 10 – IV: Das Internet der Dinge - Allgegenwärtige Informationstechnologien

(Zeit: ca. 9 UStd)

Schwerpunkte der Kompetenzentwicklung	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Information und Daten • Informatiksysteme • Informatik, Mensch und Gesellschaft • Aufbau und Wirkungsweise von Automaten • Erstellung und Analyse von Quelltexten	• Information, Daten und ihre Codierung • Erfassung, Verarbeitung und Verwaltung von Daten • Aufbau und Funktionsweise von Informatiksystemen und ihren Komponenten • Anwendung von Informatiksystemen • Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen • Chancen und Risiken bei der Nutzung von Informatiksystemen

Begriffsklärung „Internet of Things“, Funktionalität und technische Grundlagen an ausgewählten Beispielen, Aufbau und Wirkungsweise von Automaten, rechtliche Rahmenbedingungen, gesellschaftliche Akzeptanz und Auswirkungen Betriebsbesichtigung bei einem ortsansässigen Agrarmaschinenhersteller, der bereits IoT-Produkte im Herstellungsprozess nutzt.

UV 10 – V: Vertiefendes Projekt

(Zeit: ca. 18 UStd)

Schwerpunkte der Kompetenzentwicklung, Inhaltsfelder und inhaltliche Schwerpunkte sind projektabhängig.

Beispiele für Projekte:

- Erstellen eines Webauftritts mit dynamischen Teilen (evtl. unter Nutzung von JavaScript)
- Planung und Durchführung eines Programmierprojektes, Erstellung und Testen von Programmbausteinen in Gruppen
- Bearbeitung von Aufgaben aus Robotik-Wettbewerben
- Programmierung einer Quiz-App oder eines Vokabeltrainers unter Verwendung von Listen oder Arrays

- Programmierung von Mikrokontrollern mit Sensoren und Aktoren (Arduino, Raspberry PI) mit Python
- ...

Die Projektdokumentation ersetzt eine Klassenarbeit.

Spezielle Regelungen für Sekundarstufe II

Leistungsbewertung

Die Fachkonferenz Informatik orientiert sich bei der Leistungsbewertung an den im Kernlehrplan Informatik vorgegebenen Kompetenzen³. Dabei werden die Aufgabenstellungen und Kompetenzerwartungen von der EF hin zur Q2 immer komplexer und sollen zu Aufgabenstellungen auf Abiturniveau hinführen⁴.

Schriftliche Arbeiten (Klausuren)

Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse einer vorausgegangenen Unterrichtssequenz. Bei schriftlichen Arbeiten wird die Note „ausreichend minus“ analog zur Abiturbewertung ab 40 % gesetzt.

Die Dauer der Klausuren wird in Minuten angegeben.

Grundkurs:

GK	EF.I		EF.II		Q1.1		Q1.2		Q2.1		Vorabitur	Abitur
	Anz.	Dauer	Anz.	Dauer	Anz.	Dauer	Anz.	Dauer	Anz.	Dauer	Dauer	Dauer
Informatik	1	90	2	90	2	90	2	120	2	150	225	225

Leistungskurs:

LK	Q1.1		Q1.2		Q2.1		Vorabitur	Abitur
	Anz.	Dauer	Anz.	Dauer	Anz.	Dauer	Dauer	Dauer
Informatik	2	135	2	150	2	225	270	270

In der Q1 kann eine Klausur durch eine Facharbeit im Fach ersetzt werden.

³ Kernlehrplan Informatik Sekundarstufe II

(https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/KLP_GOST_Informatik.pdf) Kapitel 2

⁴ Ebenda 5 Anhang Progressionstabelle (S. 48ff).

Sonstige Mitarbeit

Zum Beurteilungsbereich „Sonstige Mitarbeit“ der Sekundarstufe II gehören alle Leistungen, die im Zusammenhang mit dem Unterricht erbracht werden mit Ausnahme der Klausuren und einer Facharbeit. Auch hier kommt es sowohl auf die Quantität als auch auf die Qualität der Leistungen an.

Zu den Leistungen der „Sonstigen Mitarbeit“ gehören beispielsweise:

- Beiträge zum Unterrichtsgespräch (Quantität und Kontinuität)
- Qualität der Beiträge (inhaltlich und methodisch)
- Eingehen auf Beiträge und Argumentationen von Mitschülerinnen und Mitschülern
- Umgang mit Arbeitsaufträgen (Hausaufgaben, Unterrichtsaufgaben,...)
- Anstrengungsbereitschaft und Konzentration auf die jeweilige Arbeit
- Darstellungsleistung bei Referaten, Plakaten oder Protokollen sowie beim Vortrag von Lösungswegen
- schriftliche Übungen gemäß §6,2 APO-SI
- selbstständiges Arbeiten, Arbeit in Gruppen sowie die Mitarbeit in Projekten sowie die Präsentation von Arbeitsergebnissen.

Auch die Teilnahme an Informatikwettbewerben kann positiv in die ‚Sonstige Mitarbeit‘ eingehen.

Die Teilnote im Beurteilungsbereich ‚Sonstige Mitarbeit‘ wird den Schülerinnen und Schülern mitgeteilt und auf Wunsch erläutert.

Schulinterner Lehrplan

Entscheidungen zum Unterricht

Hinweis: Die nachfolgend dargestellte Umsetzung der verbindlichen Kompetenzerwartungen des Kernlehrplans findet auf zwei Ebenen statt. Das **Übersichtsraster** gibt den Lehrkräften einen raschen Überblick über die laut Fachkonferenz verbindlichen Unterrichtsvorhaben pro Schuljahr. In dem Raster sind außer dem Thema des jeweiligen Vorhabens das schwerpunktmäßig damit verknüpfte Inhaltsfeld bzw. die Inhaltsfelder, inhaltliche Schwerpunkte des Vorhabens sowie Schwerpunktkompetenzbereiche ausgewiesen. Die **Konkretisierung von Unterrichtsvorhaben** führt weitere Kompetenzerwartungen auf und verdeutlicht vorhabenbezogene Absprachen, z.B. zur Festlegung auf einen Aufgabentyp bei der Lernerfolgsüberprüfung durch eine Klausur.

Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können. Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ 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“ Beispiele und Materialien, 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.

Didaktische Lernumgebung

Die Unterrichtsvorhaben in Informatik hängen in ihrer konkreten Umsetzung von der gewählten Programmierumgebung ab.

Nach der Einarbeitungsphase sollte auf eine komplexe Lernumgebung umgestellt werden. Hier bieten sich verschiedene Möglichkeiten (in alphabetischer Reihenfolge) an:

- BlueJ
- Eclipse
- IntelliJ
- NetBeans
- ...

Im Lehrwerk sind die Lehrtexte und Aufgaben so verfasst, dass jede der Umgebungen eingesetzt werden kann.

Übersichtsraster Unterrichtsvorhaben EF

Einführungsphase	
<u>UV EF – I:</u> 6 Stunden <i>Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik und die Nutzung von Informatiksystemen</i>	<u>UV EF – II:</u> 8 Stunden <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i>
<u>UV EF – III:</u> 18 Stunden <i>Algorithmische Grundstrukturen in Java</i>	<u>UV EF – IV:</u> 6 Stunden <i>Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung</i>
<u>UV EF – V:</u> 18 Stunden <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele</i>	<u>UV EF – IV:</u> 9 Stunden <i>Such- und Sortieralgorithmen</i>
<u>UV EF – VII:</u> 9 Stunden <i>Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?</i>	Gesamt: 74 Stunden

Konkretisierung der Unterrichtsvorhaben der EF

UV EF – I: *Was macht Informatik? - Einführung in die Inhaltsfelder der Informatik und die Nutzung von Informatiksystemen*

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
- Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren	- Informatiksysteme - Informatik, Mensch und Gesellschaft	- Einsatz, Nutzung und Aufbau von Informatik-systemen - Dateisystem - Internet - Wirkung der Automatisierung

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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Informatiksysteme und ihr genereller Aufbau (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)	Kapitel 1 „Was macht Informatik“ Als Anschauungsmaterial bieten sich Navigationsgeräte an
2. Der kompetente Umgang mit dem Schulnetzwerk (a) Erstellen und Anlegen von Ordnerstrukturen (b) Sortieren von Dateien und Ordern (c) Eingabe von Befehlen über Eingabeaufforderung (d) Einzelrechner und Netzwerk (e) Sicherheit und Datenschutz		Kapitel 1 „Was macht Informatik“ Interview mit dem Netzwerk-administrator, Benutzer- und Datenschutzbestimmungen der Schule

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Modellieren • Implementieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache

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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Identifikation von Objekten und Klassen (a) An einem lebensweltnahen Bei-spiel werden Objekte und Klassen im Sinne der objektorientierten Modellierung eingeführt. (b) Objekte werden durch Objekt-diagramme, Klassen durch Klassen-diagramme 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 Programmiersprache, auch unter Nutzung dokumentierter Klassenbibliotheken (I).	Kapitel 2 „Einführung in die Objektorientierung“ (2.1) Objektorientierte Modellierung
2. Der kompetente Umgang mit dem Schulnetzwerk (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

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren • Implementieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen

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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Algorithmen (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	Kapitel 3 „Algorithmen“ (3.1) Wiederholungen (3.2) Bedingte Anweisungen (3.3) Logische Funktionen (3.4) 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	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 4 „Variablen und Methoden“ (4.1) lokale Variablen (4.2) Methoden (4.3) Attribute
--	---	---

UV EF – IV: *Das ist die digitale Welt! – Einführung in die Grundlagen, Anwendungsgebiete und Verarbeitung binärer Codierung*

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
- Kommunizieren und Kooperieren - Darstellen und Interpretieren - Argumentieren	- Informatiksysteme - Informatik, Mensch und Gesellschaft	- Binäre Codierung und Verarbeitung - Besondere Eigenschaften der digitalen Speicherung und Verarbeitung von Daten

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.

Sequenzierung des Unterrichtsvorhabens:

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 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)	<i>Exkurs „Analog und 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 Binärrechners		<i>Exkurs „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>Exkurs „Arbeitsweise eines Computers“</i>

UV EF – V: *Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand lebensnaher Anforderungsbeispiele*

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Kommunizieren und Kooperieren • Darstellen und Interpretieren • Argumentieren • Modellieren und Implementieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung einfacher Algorithmen

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.

Sequenzierung des Unterrichtsvorhabens:

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 Schülerinnen und Schüler • 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),	Kapitel 6 „Klassenentwurf“ (6.1) Von der Realität zum Programm (6.2) Objekte identifizieren (6.3) Klassen und Beziehungen entwerfen
2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden		Kapitel 6 „Klassenentwurf“ (6.4) Klassen und Beziehungen implementieren (6.5) Vererbung

(b) Festlegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung	• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M),	
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.	• 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),	Kapitel 6 „Klassenentwurf“ (6.4) Klassen und Beziehungen implementieren (6.5) Vererbung
4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert	• analysieren und erläutern einfache Algorithmen und Programme (A) • modifizieren einfache Algorithmen und Programme (I),	Kapitel 6 „Klassenentwurf“ (6.5) Vererbung
5. Softwareprojekt (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	• 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 8 „Softwareprojekte“ (8.1) Softwareentwicklung (8.2) Oberflächen

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Algorithmen • Daten und ihre Strukturierung	• Algorithmen zum Suchen und Sortieren • Analyse, Entwurf und Implementierung einfacher Algorithmen • Objekte und Klassen

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 Sortierv Verfahren beurteilt.

Sequenzierung des Unterrichtsvorhabens:

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 Schülerinnen und Schüler • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D)	Kapitel 7 „Sortieren und Suchen auf Feldern“ (7.1) Das Feld – Eine Sammlung von Daten
2. Explorative Erarbeitung von Suchverfahren (a) Erkundung von Strategien für das Suchen auf unsortierten Daten, auf sortierten Daten und	• entwerfen einen weiteren Algorithmus zum Sortieren (M)	Kapitel 7 „Sortieren und Suchen auf Feldern“ Projekteinstieg 1: Suchen (7.2) Suchen mit System

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)	• 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 Sortialgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs (e) Analyse eines weiteren Sortieralgorithmus (sofern nicht in (a) bereits geschehen)		Kapitel 7 „Sortieren und Suchen auf Feldern“ Projekteinstieg 2: Sortieren (7.3) Ordnung ist das halbe Leben!? – Sortieren • Sortieren • Selection Sort • Insertion Sort • Bubble Sort

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

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Kommunizieren und Kooperieren • Darstellen und Interpretieren • Argumentieren	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Geschichte der automatischen Datenverarbeitung • Wirkungen der Automatisierung • Digitalisierung

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.

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Schriftzeichen, Rechenmaschine, Computer (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.	Die Schülerinnen und Schüler • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A),	<i>Exkurs „Geschichte der Informatik“</i>
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.	• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K)	• <i>Exkurs „Geschichte der Informatik“</i>

Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase

Qualifikationsphase 1	
<u>UV Q1 – I:</u> 8 Stunden Wiederholung und Vertiefung der objektorientierten Modellierung anhand einer kontextbezogenen Problemstellung	<u>UV Q1 – II:</u> 20 Stunden Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen
<u>UV Q1 – III:</u> 16 Stunden Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen	<u>UV Q1 – IV:</u> 20 Stunden Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten
<u>UV Q1 – V:</u> 10 Stunden Sicherheit und Datenschutz in Netzstrukturen	Gesamt: 74 Stunden
Qualifikationsphase 2	
<u>UV Q2 – I:</u> 24 Stunden Modellierung und Implementierung von Anwendungen mit dynamischen nicht - linearen Datenstrukturen	<u>UV Q2 – II:</u> 20 Stunden Endliche Automaten und formale Sprachen
<u>UV Q2 – III:</u> 12 Stunden Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit	<u>UV Q2 – IV:</u> Wiederholung und Vertiefung ausgewählter Kompetenzen und Inhalte des ersten Jahres der Qualifikationsphase
Gesamt: 56 Stunden	

Konkretisierung der Unterrichtsvorhaben der Q1 (Grundkurs)

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).

UV Q1 – I: Wiederholung und Vertiefung der objektorientierten Modellierung anhand einer kontextbezogenen Problemstellung

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren und Modellieren • Darstellen und Interpretieren • Implementieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten • Informatiksysteme, Mensch und Gesellschaft	• Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Syntax und Semantik einer Programmiersprache • Nutzung von Informatiksystemen

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. An-hand 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 Schülerinnen und Schüler Klassenentwürfe.

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

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung (a) Sichtweise der objektorientierten Informatik auf die Welt Analyse einer Problemstellung (b) OOP als informatikspezifische Modellierung der Realität (c) Schritte der Softwareentwicklung	Die Schülerinnen und Schüler - analysieren und erläutern objektorientierte Modellierungen (A), - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - 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),	(1.2) Die Welt ist voller Objekte Projekteinstieg: Klassenentwurf – step by step Beispiel: Zoohandlung
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		(1.3) Gut geplant – Klassenentwurf (1.4) Vererbungshierarchien nutzen
3. Mensch und Technik (a) Verantwortung von Informatikern (b) Automatisierung des Alltags durch Informatik		Die digitale Welt 001 - Mensch und Technik
4. Übung und Vertiefung der OOM / OOP		Prüfungsvorbereitung

	• interpretieren Fehlermeldungen und korrigieren den Quellcode (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).	
--	---	--

UV Q1 – II: Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Modellieren • Implementieren • Darstellen und Interpretieren • Argumentieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Syntax und Semantik einer Programmiersprache • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten

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.

Sequenzierung des Unterrichtsvorhabens:

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 Schülerinnen und Schüler • erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A) • konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M) • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M)	(2.1) + (2.2) Anforderungen an eine Datenstruktur Speichern mit Struktur neue Wiederholung: Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)
2. Die Datenstruktur Schlange (a) Modellierung und Implementierung der Verknüpfung von Objekten (b) Generische Typen, Trennung von Verwaltung und Inhalt		(2.3) - Wer zuerst kommt - Objekte miteinander verketteten - Verwaltung und Inhalt - Funktionen der Queue

dynamischen Datenstrukturen. (c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden (d) Funktionalität der Schlange unter Verwendung der Klasse Queue; Erschließen der Standardoperationen (e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse Queue	• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) • testen und korrigieren Computerprogramme (I) • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) • stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) • 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)	Aufgaben: - Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist. 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 gewissen Menge an Verstößen vorliegt oder Autos kontrolliert wurden.
3. Die Datenstruktur Stapel (a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden (b) Funktionalität der Klasse Stapel unter Verwendung der Klasse Stack (c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse Queue (d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen Queue, Stack und Array (Palindrom))	• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) • testen und korrigieren Computerprogramme (I) • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) • stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) • 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)	(2.4) - Daten gut abgelegt – Stapel - Funktionen der Datenstruktur Stapel Aufgaben: - Standardoperationen / Basiskompetenz (Stapel Münzen/ CDs) zur Umsetzung der gegebenen Funktionen der Klasse Stack - 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 Queue als auch Stack verwendet werden. - Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.

4. Die Datenstruktur Liste (a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu Stack/Queue, zusätzliche Fähigkeiten der Klasse List) (b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse List (c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse List (d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Stack, Queue, List)		(2.5) - Flexibel für alle Fälle – (die) lineare Liste - Funktionen der Datenstruktur Liste Aufgaben: - LEDs - Textzeilen verarbeiten
5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen		Prüfungsvorbereitung

UV Q1 – III: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Darstellen und Interpretieren • Modellieren • Implementieren • Kommunizieren und Kooperieren	• Algorithmen • Formale Sprachen und Automaten	• Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatischen Kontexten • Syntax und Semantik einer Programmiersprache

Leitfragen: Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortialgorithmen 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 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 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 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.

Sequenzierung des Unterrichtsvorhabens:

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 Schülerinnen und Schüler • analysieren und erläutern Algorithmen und Programme (A), • modifizieren Algorithmen und Programme (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). • implementieren und erläutern iterative und	(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

3. Sortieren auf Listen und Arrays (a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste (b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array (c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf	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 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), • testen Programme systematisch anhand von Beispielen (I),	(3.4) Sortieren – iterativ und rekursiv
--	--	---

UV Q1 – IV: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
- Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren - Kommunizieren und Kooperieren	- Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatik, Mensch und Gesellschaft	- Datenbanken - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Sicherheit

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 wird 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 (soweit nötig) normalisiert.

Sequenzierung des Unterrichtsvorhabens:

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, Daten-satz, 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,	Die Schülerinnen und Schüler • erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), • analysieren und erläutern eine Datenbank-modellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • bestimmen Primär- und Sekundärschlüssel (M), • ermitteln für anwendungsbezogene Problem-stellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • überführen Datenbankschemata in	Kapitel 7 „Datenbanken“ (7.1) Wissen speichern und verwalten – Datenbanksysteme (7.2) Daten anordnen mit Tabellen Beispiel: Buchhandlung Redundanzen, Anomalien und Inkonsistenzen Projekteinstieg: Tabellen (7.3) Daten filtern mit SQL (7.4) Komplexe Filter Aufgaben z.B. Bundesligatabellen, Videocenter oder Schulbuchausleihe

geschachtelte Select-Ausdrücke) (c) Vertiefung an einem weiteren Datenbankbeispiel	vorgegebene Normalformen (M), • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), • ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), • stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), • überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	
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-Diagrammes Erläuterung und Erweiterung einer Datenbankmodellierung (b) Entwicklung eines relationalen Modells aus einem Datenbank-entwurf 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 Beispiel: Online-Buchhandel Datenanalyse und Entwurf (7.6) Umsetzung des ER-Modells Entitätsmengen m:n-Beziehungen 1:n-Beziehungen 1:1-Beziehungen Wiederaufgriff des Projekteinstiegs (7.7) Datenbanken verbessern durch Normalformen
3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken		Prüfungsvorbereitung

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Einzelrechner und Rechnernetzwerke • Sicherheit • Nutzung von Informatiksystemen, Wirkungen der Automatisierung

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

Vorhabenbezogene Konkretisierung:

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

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Daten in Netzwerken und Sicherheitsaspekte in Netzen sowie beim Zugriff auf Datenbanken (a) Beschreibung eines Datenbankzugriffs im Netz anhand eines Anwendungskontextes und einer Client-Server-Struktur zur Klärung der Funktionsweise eines Datenbankzugriffs (b) Netztopologien als Grundlage von Client-Server-Strukturen und TCP/IP-Schichtenmodell als Beispiel für eine Paketübermittlung in einem Netz (c) Vertraulichkeit, Integrität, Authentizität in Netzwerken sowie symmetrische und	Die Schülerinnen und Schüler • beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des	Materialien: zum Beispiel: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken Materialien aus dem Buch Informatik 2 (6.2) Ohne Protokoll läuft nichts – Netzwerke (6.3) Einer für alle – Client-Server Strukturen

asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen	Einsatzes von Informatiksystemen, die Sicherheit von Informatiksystemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A),	Die digitale Welt 100 – Kryptologie
2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	• untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Interessenlagen (A), • nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D).	Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1 5 - Datenschutz beim Videocenter, Materialblatt-Datenschutzgesetz

Konkretisierung der Unterrichtsvorhaben der Q2 (Grundkurs)

UV Q2 – I: Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Darstellen und Interpretieren • Modellieren • Implementieren • Kommunizieren und Kooperieren	• Daten und ihre Strukturierung • Algorithmen • Formale Sprachen und Automaten	• Objekte und Klassen • Analyse, Entwurf und Implementierung von Algorithmen • Algorithmen in ausgewählten informatorischen Kontexten • Syntax und Semantik einer Programmiersprache

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.

Sequenzierung des Unterrichtsvorhabens:

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 Schülerinnen und Schüler • 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). • beurteilen die syntaktische Korrektheit und die	(5.1) Spielen mit Struktur – Baumstrukturen Projekteinstieg 1: Spielbäume
2. Binäre Bäume (a) rekursiver Aufbau eines binären Baums		(5.2) Zwei Nachfolger sind genug! – Binäre Bäume

(b) Traversierungen (pre-, in-, postorder) (c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm) (d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)	Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I),	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)	Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I),	(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	Funktionalität von Programmen (A), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und „Teilen und Herrschen“ (M), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • modifizieren Algorithmen und Programme (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I),	Prüfungsvorbereitung

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Modellieren • Darstellen und Interpretieren • Kommunizieren und Kooperieren	• Endliche Automaten und formale Sprachen	• Endliche Automaten • Grammatiken regulärer Sprachen • Möglichkeiten und Grenzen von Automaten und formalen 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.

Sequenzierung des Unterrichtsvorhabens:

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 DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere	Die Schülerinnen und Schüler - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - 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 endlicher 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) - 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)	Vom realen Automaten zum Modell Projekteinstieg: Schatzsuche - Der Mealy-Automat - Der erkennende endliche Automat - Wort und Sprache Beispiele: - Cola-Automat, - Geldspielautomat, - Roboter, Zustandsänderung eines Objekts „Auto“, - Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen (Download Q2-II.1)
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. (c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden (d) Entwicklung regulärer Grammatiken zu endlichen Automaten	(siehe oben)	Grammatiken regulärer Sprachen

3. Übungen und Vertiefungen / Grenzen endlicher Automaten (a) Verwendung endlicher Automaten und Grammatiken regulärer Sprachen	• zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf	Beispiele: Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$ Prüfungsvorbereitung
4. Projekteinstieg (a) Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben (Schatzinsel)		

UV Q2 – III: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zentrale Kompetenzen	Inhaltsfelder	Inhaltliche Schwerpunkte
• Argumentieren • Kommunizieren und Kooperieren	• Informatiksysteme • Informatik, Mensch und Gesellschaft	• Einzelrechner und Rechnernetzwerke • Grenzen der Automatisierung

Vorhabenbezogene Konkretisierung:

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

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

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Kapitel und Materialien
1. Von-Neumann-Architektur und die Ausführung maschinennaher Programme (a) prinzipieller Aufbau einer von Neumann-Architektur mit CPU, Rechenwerk, Steuerwerk, Register und Hauptspeicher (b) einige maschinennahe Befehlen und ihre Repräsentation in einem Binär-Code, der in einem Register gespeichert werden kann (c) Analyse und Erläuterung der Funktionsweise eines einfachen maschinennahen Programms	Die Schülerinnen und Schüler • 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).	Beispiel: Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 – Von-Neumann-Architektur und maschinennahe Programmierung (Download Q2-III.1)
2. Grenzen der Automatisierbarkeit (a) Vorstellung des Halteproblems (b) Unlösbarkeit des Halteproblems (c) Beurteilung des Einsatzes von Informatiksystemen hinsichtlich prinzipieller Möglichkeiten und prinzipieller Grenzen		Beispiel: Halteproblem Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 - Halteproblem (Download Q2-III.2) Die digitale Welt 100 – Kryptologie Prüfungsvorbereitung

Facharbeit

Im Fach Informatik können in der Q1 Facharbeiten geschrieben werden. Diese können sowohl die Entwicklung einer eigenen Software als auch allgemeine wissenschaftliche Auseinandersetzung mit dem aktuellen Stand der Forschung bei einer bestimmten Fragestellung als Thema haben.

Außerunterrichtliches

Exkursionen

Zurzeit gibt es keine festen Exkursionen zu vorhandenen IT-Unternehmen. Jedoch gibt es Firmen, die mit ihrem praktischen Know How den Unterricht am OHG an die Praxis ankoppeln.

Wettbewerbe

Das OHG nimmt jedes Jahr mit allen Schülerinnen und Schülern, die das Fach Informatik, ENW oder informatische Grundbildung belegt haben am Informatik-Biber Wettbewerb teil. Je nach Angebot kommen weitere Wettbewerbe wie der Wettbewerb der Kölnischen Rundschau, der AntMe-Wettbewerb des MINT-Netzwerks unter Federführung der FHDW in Bergisch Gladbach und andere Wettbewerbe hinzu.

Weitere Fachvereinbarungen

Sammlung

Zur Sammlung gehören neben einigen Calliopes für den ENW Unterricht und ein Raspberry. Weitere Hardware wie Arduinos, Raspberries und Roboter bekommt das Fach Informatik auf kollegialem Weg von dem Fach Technik.

Fachschaftsordner

Die Fachschaft sammelt Ihre schriftlichen, digitalen Unterrichtsmaterialien im Fachschaftsordner in OneDrive.