Schulinternen Lehrplan
zum Kernlehrplan für die gymnasiale Oberstufe

im Fach Informatik

Stand: 09.11.2021

Inhaltsverzeichnis

1 Vorstellung des Informatikunterrichts am Rupert-Neudeck-Gymnasium 3
1.1 Vorstellung der Schule und der Fachgruppe 3
1.2 Informatikunterricht in der Gymnasialen Oberstufe 3
2 Entscheidungen zum Unterricht 4
2.1 Unterrichtsvorhaben 4
2.1.1 Einführungsphase 4
2.1.2 Qualifikationsphase 10
2.2 Grundsätze der Leistungsbewertung und Leistungsrückmeldung 23
2.2.1 Vorbemerkungen 23
2.2.2 Bewertungsbereich Sonstige Mitarbeit 24
2.2.3 Klausuren 25
2.2.4 Grundsätze der fachmethodischen und fachdidaktischen Arbeit 26
3 Qualitätssicherung und Evaluation 27

1 Vorstellung des Informatikunterrichts am Rupert-Neudeck-Gymnasium

1.1 Vorstellung der Schule und der Fachgruppe

Das Rupert-Neudeck-Gymnasium der Gemeinde Nottuln liegt zentral im ländlichen Raum des westlichen Münsterlands. Es wurde im Jahr 1991 als Gymnasium Nottuln gegründet und änderte im Jahr 2017 den Namen in Rupert-Neudeck-Gymnasium um. Es handelt sich um ein zwei- bis dreizügiges Gymnasium. Etwas über 400 Schülerinnen und Schüler besuchen das Gymnasium. Sie werden von 46 Lehrerinnen und Lehrern unterrichtet, wobei das Fach Informatik von drei Kolleginnen und Kollegen unterrichtet wird. Seit dem Schuljahr 2016/17 wird das Fach am Rupert-Neudeck-Gymnasium angeboten. In zwei voll ausgestatteten Rechnerräumen mit 25 und 30 Arbeitsplätzen kann der Unterricht durchgeführt werden. Seit dem Schuljahr 2019/20 werden die Schülerinnen und Schüler sukzessive mit iPads ausgestattet.

Jgst. 5	Informatik (halbjährig 2 Wochenstunden)
Jgst. 6	Informatik (halbjährig 2 Wochenstunden)
Jgst. 9	Informatik WP II (ganzjährig 3 Wochenstunden)
Jgst. 10	Informatik WP II (ganzjährig 3 Wochenstunden)
EF	Grundkurs Informatik (3-stündig)
Q1	Grundkurs Informatik (3-stündig)
Q2	Grundkurs Informatik (3-stündig)
Abitur	mdl. oder schriftlich im Grundkurs

Informatische Inhalte und Fragestellungen werden am Rupert-Neudeck-Gymnasium im Wahlpflichtbereich II der Jahrgangsstufe 8 und 9 mit je drei Wochenstunden und in der Gymnasialen Oberstufe im Grundkurs behandelt.

1.2 Informatikunterricht in der Gymnasialen Oberstufe

Die Vorgaben für das Zentralabitur NRW regeln im Wesentlichen den Ablauf des Informatikunterrichts in der Gymnasialen Oberstufe. Als Entwicklungsumgebungen für den Einstieg in die Objektorientierte Programmierung werden BlueJ und Greenfoot als auch der JavaEditor genutzt.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

2.1.1 Einführungsphase

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

Die Schülerinnen und Schüler

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

Unterrichtsvorhaben EF-I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Was ist Informatik? a) Bereiche der Informatik – Theoretische Informatik – Praktische Informatik – Technische Informatik – Angewandte Informatik b) Informatik und ihre Auswirkungen auf die heutige Gesellschaft	Die Schülerinnen und Schüler – beschreiben und erläutern die Grundideen fundamentaler informatischer Konzepte (Inhaltsfelder) (A) – bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A).	Material: net-schulbuch.de - „Was ist Informatik“ - Als Anschauungsmaterial bieten sich Navigationsgeräte an. net-schulbuch.de - „Arbeitswandel“
2. Information, deren Kodierung und Speicherung a) Informatik als Wissenschaft der Verarbeitung von Informationen b) Darstellung von Informationen in Schrift, Bild und Ton. c) Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner d) Vereinbarung von Richtlinien zur Datenspeicherung auf den Schul-rechnern (z.B. Ordnerstruktur, Dateibezeichner usw.)	Die Schülerinnen und Schüler – nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D), – nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	*Beispiel:* Textcodierung Kodierung und Dekodierung von Texten mit dem ASCII-Code und Unicode. *Beispiel:* Bildkodierung Kodierung von Bildinformationen in Raster- und Vektorgrafiken

3. Prinzipieller Aufbau und Arbeitsweise eines Rechners a) EVA Prinzip b) von-Neumann-Architektur Speichern von Daten mit informatischen Systemen am Beispiel der Schulrechner und Simulationsprogrammen → Netzwerk	– beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).	*Material*: Demonstrationshardware - Bestandteile eines Rechners identifizieren und kategorisieren - „Die digitale Welt 011 – Der Von-Neumann-Rechner“ (Informatik 1, Schöningh) - net-schulbuch.de - Einzelrechner - Schulnetzwerk

Unterrichtsvorhaben EF-II

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

Zeitbedarf: ca. 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Identifikation von Objekten a) Am Beispiel lebensweltnaher Objekte wird die Objektorientierten Modellierung eingeführt. b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen. c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Objektsorte bzw. Objektklasse zusammengefasst.	Die Schülerinnen und Schüler – ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), – modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), – implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), – stellen den Zustand eines Objekts dar (D).	Material: z.B. Schüleretui Schülerinnen und Schüler betrachten einen Werkzeugkoffer oder Vogelschwarm als Menge gleichartiger Objekte, die in einer Klasse mit Attributen und Methoden zusammengefasst werden können. Materialien: Ergänzungsmaterialien zum Lehrplannavigator - Allgemeine Objektorientierung (Download EF-II.1)
2. Analyse von Klassen didaktischer Lernumgebungen a) Objektorientierte Programmierung als modularisiertes Vorgehen b) Teilanalyse der Klassen der didaktischen Lernumgebungen BlueJ und der Grafikbibliothek „Shapes and Sprites“		Materialien: - net-schulbuch.de Dokumentation der didaktischen Bibliothek sas. - Drei Grafiken - Zwei Vierecke
3. Implementierung erster Linearer Programme a) Grundaufbau einer Java-Klasse b) Dokumentation einer Klasse c) Konstruktoren und Variablen d) Deklaration und Initialisierung von Objekten e) Variablen f) Methodenaufrufe mit Parameterübergabe zur Manipulation von Objekteigenschaften (z.B. Farbe, Position, Drehung)		Beispiele: - BewegteVierecke und MuellSammeln - Ballon aufblasen SuS erstellen ein Programm, das mit Hilfe von geometrischen Objekten der „Shapes and Sprites-Umgebung“ verschiedenste Bilder zeichnet. - Bildergalerie - Kirchenfenster

Unterrichtsvorhaben EF-III

Thema: Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java anhand von einfachen Animationen

Zeitbedarf: ca. 18 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. **Bewegungsanimationen am Beispiel einfacher grafischer Objekte (SaS-Objekte)** a) Kontinuierliche Verschiebung eines SaS-Objekts mit Hilfe einer Schleife (while-Schleife) b) Tastaturabfrage zur Realisierung einer Schleifenbedingung für eine Animationsschleife c) Mehrstufige Animationen mit mehreren sequenziellen Schleifen d) Berechnung von Abständen zwischen SaS-Objekten mit Hilfsvariablen e) Meldungen zur Kollision zweier SaS-Objekte mit Hilfe von Abstandsberechnungen und Verzweigungen (IF-Anweisungen)	Die Schülerinnen und Schüler – analysieren und erläutern einfache Algorithmen und Programme (A), – entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), – ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), – modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), – ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), – ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), – modifizieren einfache Algorithmen und Programme (I), – implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), – implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), – implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), – testen Programme schrittweise anhand von Beispielen (I), – interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	Material: PAP-Designer net-schulbuch.de - Kontrollstrukturen Beispiel: Ball mit Hindernis Beispiel: Mond um die Erde SuS realisieren mit Objekten der Sas-Umgebung eine Simulation, bei der sich der Mond um die Erde bewegt. Beispiel: Reaktionstest SuS realisieren mit Objekten der Sas-Umgebung ein Spiel, bei der der Spieler mit der Maus auf einen spontan erscheinenden Kreis klicken sollen. Beispiel: Pfeilwurf SuS realisieren mit Objekten der Sas-Umgebung ein Spiel, bei dem ein Pfeil über den Bildschirm bewegt wird und auf eine Zielscheibe trifft.
2. Erstellen und Verwalten größerer Mengen einfacher grafischer Objekte (SaS-Objekte) a) Erzeugung von Objekten mit Hilfe von Zählschleifen (FOR-Schleife) b) Verwaltung von Objekten in eindimensionalen Feldern (Arrays) c) Animation von Objekten, die in eindimensionalen Feldern (Arrays) verwaltet werden d) Vertiefung: Verschiedene Feldbeispiele		Material: net-schulbuch.de - Kontrollstrukturen Beispiel: z.B. Münzsuchroboter SuS realisieren mit Objekten der Sas-Umgebung eine Simulation, bei dem ein Roboter systematisch ein Feld nach einer Münze absuchen soll. Beispiel: z.B. Staubsauger SuS realisieren mit Objekten der Sas-Umgebung eine Simulation, bei der ein Staubsauger über einen Teppich bewegt wird und dabei Staubkörner, die in einem Array verwaltet werden, aufsaugt.
3. Modellierung und Animation komplexerer grafisch repräsentierbarer Objekte a) Modellierung eines Simulations-programms mit eigenen Klassen, die sich selbst mit Hilfe von einfachen SaS-Objekten zeigen mit Hilfe eines Implementationsdiagramms		Material: PAP-Designer net-schulbuch.de - Kontrollstrukturen

b) Implementierung eigener Methoden mit und ohne Parameterübergabe c) Realisierung von Zustandsvariablen d) Thematisierung des Geheimnisprinzips und des Autonomitätsprinzips von Objekten e) Animation mit Hilfe des Aufrufs von selbstimplementierten Methoden f) Erarbeitung des Unterschieds zwischen Aufträgen und Abfragen g) Vertiefung: Weitere Projekte		Beispiel: Breakout SuS realisieren mit Objekten der sas-Umgebung ein Spiel, bei dem ein Ball mit einem Schläger immer wieder ins Spielfeld zurückgespielt werden soll und dabei eine Reihe von Würfeln ab trifft. Beispiel: z.B. BälleKlicken SuS realisieren mit Objekten der sas-Umgebung ein Spiel, bei dem willkürlich herunterfallende Bälle mit der Maus angeklickt werden sollen. - Konsolenprogramme zum vertieften Üben der Kontrollstrukturen z.B. Ticketautomat

Unterrichtsvorhaben EF-IV

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Darstellungen und Simulationen

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Grundlagen der OOP a) Erarbeitung der Bedeutung von Klassen, Objekten und Methoden b) Darstellung von Klassen und Objekten in UML-Notation	Die Schülerinnen und Schüler – analysieren und erläutern eine objektorientierte Modellierung (A), – stellen die Kommunikation zwischen Objekten grafisch dar (M), – ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), – modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), – ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), – ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), – modellieren Klassen unter Verwendung von Vererbung (M), – implementieren Klassen in einer Programmiersprache	Material: net-schulbuch.de - Modellierung von Klassen Beispiel: Digitaluhr und Windpark
2. Objektbeziehungen modellieren und umsetzten a) Modellierung des Spiels mit Hilfe eines Implementationsdiagramms b) Dokumentation der Klassen des Projekts c) Implementierung einer Oberklasse d) Das Prinzip der Vererbung - Modellieren und implementieren Unterklassen (Implementations-diagramm und Quellcode) e) Verallgemeinerung und Reflexion des Prinzips der Vererbung am Beispiel der Spezialisierung e) Darstellung der Kennt- und Ist-Beziehung in Form von Klassendia-grammen		Material: net-schulbuch.de - Implementation von Klassen Beispiel: z.B. Ufospiel SuS entwickeln die Simulation eines Ufos, das Asteroiden ausweichen soll mit denen eine Kollision möglich ist. - Vererbung Beispiel: Turner SuS entwickeln eine Simulation von Turnern, bei der unterschiedliche Turner unterschiedliche Bewegungen durchführen können. z.B. Schrankwand oder Häuser - Greenfoot Szenarien für Greenfoot, z.B.: „Krabbe am Strand“

f) Spezialisierung einer Klasse zu Unterklassen mit verschiedenen Fähigkeiten durch Überschreiben von Methoden g) Polymorphie und das Prinzip der späten Bindung h) Abstrakte Oberklassen	auch unter Nutzung dokumentierter Klassenbibliotheken (I), – testen Programme schrittweise anhand von Beispielen (I), – interpretieren Fehlermeldungen und korrigieren den Quellcode (I), – modifizieren einfache Algorithmen und Programme (I), – stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), – dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).	

Unterrichtsvorhaben EF-V

Thema: Such- und Sortieralgorithmen anhand kontextbezogener Beispiele

Zeitbedarf: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	**Zu entwickelnde Kompetenzen**	**Beispiele, Medien Materialien**
1. Lineare und Binäre Suche auf sortierten Daten a) Suchaufgaben im Alltag und im Kontext informatischer Systeme b) Evtl. Simulationsspiel zum effizienten Suchen mit binärer Suche c) Effizienzbetrachtungen zur binären Suche	Die Schülerinnen und Schüler – beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), – entwerfen einen weiteren Algorithmus zum Sortieren (M), – analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).	Material: - Telefonbuch - z.B. Einführungsspiel zur Binären Suche - net-schulbuch.de Suchen und Sortieren
2. Explorative Erarbeitung eines Sortierverfahrens a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch, Bundesligatabelle, usw.) b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler		PAP-Designer Beispiel: Sortieren mit Karten SuS bekommen die Aufgabe Karten zu sortieren. Materialien: - Kartenspiele - net-schulbuch.de Sortierverfahren - Computer science unplugged – Sorting Algorithms, URL: www.csunplugged.org/sorting-algorithms abgerufen: 09. 05. 2018
3. Systematisierung von Algorithmen und Effizienzbetrachtungen a) Formulierung (falls selbst gefunden) oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Sortieren durch Vertauschen, Sortieren durch Auswählen)		Beispiel: z.B. Länderkarten SuS visualisieren die Abläufe der Sortieralgorithmen Material: net-schulbuch.de - Sortierverfahren

b) Anwendung von Sortier-algorithmen auf verschiedene Beispiele c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche d) Variante des Sortierens durch Auswählen (Nutzung eines einzigen oder zweier Felder bzw. lediglich eines einzigen zusätzlichen Ablage-platzes oder mehrerer neuer Ablage-plätze) e) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatz-bedarfs f) Analyse eines weiteren Sortier-algorithmus (sofern es nicht bereits geschehen ist)		

Unterrichtsvorhaben EF-VI

Thema: Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes

Zeitbedarf: ca. 8 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1 Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler a) Mögliche Themen zur Erarbeitung in Kleingruppen, sofern noch nicht im Unterrichtsvorhaben I thematisiert: • „Der Weg von ersten Rechenhilfen bis zum modernen Minicomputer.“ • „Eine kleine Geschichte der Krypto-graphie: von Caesar zur Enigma“ • „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ • Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ • „Von Logik-Operationen und Aussagenlogik zu Integrierten Schaltungen“ • Internet – Geschichte, Funktionsweise des Internets, Dienste des Internets und Entstehung einer Internetseite b) Vorstellung und Diskussion durch Schülerinnen und Schüler	Die Schülerinnen und Schüler – bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), – erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), – stellen ganze Zahlen und Zeichen in Binärcodes dar (D), – interpretieren Binärcodes als Zahlen und Zeichen (D), – nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).	Material: Internetrecherche net-schulbuch.de - Geschichte der Informatik ggf. Auswirkung der Kryptologie im 2. Weltkrieg auf die Entwicklung der Informatik

2. Betrachtung des Themas Datenschutz Selbstentdeckendes Erkunden: a) Umfang von Datensammlungen, b) vorgehen bei Datensammlungen, c) wirtschaftlichen Interessen bei Datensammlungen d) Motivation der Akteure Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler		Material: Onlinespiel *DataDealer*

2.1.2 Qualifikationsphase

Unterrichtsvorhaben Q1-I

Thema: Wiederholung der objektorientierten Modellierung und Programmierung

Zeitbedarf: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Wiederholung von Objekten und Klassen und deren Darstellung mit Hilfe von Grafischen Benutzeroberflächen • Konstruktion Grafischer Benutzeroberflächen	Die Schülerinnen und Schüler • analysieren und erläutern objektorientierte Modellierungen (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • wenden eine didaktisch orientierte Entwicklungsumgebung zur	Material: net-schulbuch.de - Vertiefung OOP / Benutzeroberflächen Beispiele: - Nimm-Spiel - Währungsrechner - Vokabeltrainer
2. Wiederholung und Erweiterung der objektorientierten Modellierung und Programmierung durch Analyse und Erweiterung eines kontextbezogenen Beispiels a) Analyse der Problemstellung b) Analyse der Modellierung (Implementationsdiagramm) c) Erweiterung der Modellierung im Implementationsdiagramm (Vererbung, abstrakte Klasse) d) Kommunikation zwischen mindestens zwei Objekten (grafische Darstellung) e) Dokumentation von Klassen f) Implementierung der Anwendung oder von Teilen der Anwendung		Material: - net-schulbuch.de - Schöningh: Informatik 2: Beispiele: - Personalverwaltung - Oberstufenschüler - Lastwagen - Bankkonto - Lohnbüro - Gebäude

	Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • stellen die Kommunikation zwischen Objekten grafisch dar (D).	

Unterrichtsvorhaben Q1-II

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

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	**Zu entwickelnde Kompetenzen**	**Beispiele, Medien Materialien**
1. Die Datenstruktur Schlange im Anwendungskontext unter Implementierung einer Klasse Warteschlange a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen b) Erarbeitung der Funktionalität einer Warteschlange c) Modellierung und Implementierung der Datenstruktur Warteschlange	Die Schülerinnen und Schüler • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der	Beispiele: - Gästewarteschlange - Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) Patientenwarteschlange (jeder kennt seinen Nachfolger bzw. alternativ: seinen Vorgänger) Sobald ein Patient in einer Arztpraxis eintrifft, werden sein Name und seine Krankenkasse erfasst. Die Verwaltung der Patientenwarteschlange geschieht über eine Klasse, die hier als Wartezimmer bezeichnet wird. Wesentliche Operationen sind das „Hinzufügen“ eines Patienten und das „Entfernen“ eines Patienten, wenn er zur Behandlung gerufen wird. Die Simulationsanwendung stellt eine GUI zur Verfügung, legt ein Wartezimmer an und steuert die Abläufe. Wesentlicher Aspekt des Projektes ist die Modellierung des Wartezimmers mit Hilfe der Klasse Queue. Anschließend wird der Funktionsumfang der Anwendung erweitert: Patienten können sich zusätzlich in die Warteschlange zum Blutdruckmessen einreihen.

	Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D).	Objekte werden von zwei Schlangen verwaltet. Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 – Warteschlange Material: Lineare Datenstrukturen - net-schulbuch.de - Schöningh: Informatik 2
2. Die Datenstruktur Stapel im Anwendungskontext unter Nutzung der Klasse Stack a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen b) Erarbeitung der Funktionalität der Klasse Stack c) Modellierung und Implementierung der Anwendung unter Verwendung eines oder mehrerer Objekte der Klasse Stack		Material: net-schulbuch.de - Lineare Datenstrukturen Beispiel: Heftstapel In einem Heftstapel soll das Heft einer Schülerin gefunden werden. oder Beispiel: Wortumkehrer In einem Stapel werden Buchstaben entsprechend des übergebenen Strings gestapelt. Der String soll nun umgekehrt ausgegeben werden. oder Beispiel: Restaurant Überprüfung mithilfe eines Stapels, ob vorbereitete Geschirrstapel zur entsprechenden Kundenschlange passt. Material: - Schöningh: Informatik 2
3. Die Datenstruktur lineare Liste im Anwendungskontext unter Nutzung der Klasse List a) Erarbeitung der Vorteile der Klasse List im Gegensatz zu den bereits bekannten linearen Strukturen b) Modellierung und Implementierung einer kontextbezogenen Anwendung unter Verwendung der Klasse List.		Material: net-schulbuch.de - Lineare Datenstrukturen Beispiel: Torjägerliste In einer Torjägerliste werden nach und nach Torschützen in eine Liste aufgenommen und entsprechend ihrer geschossenen Tore in eine Rangliste eingeordnet. Die neuen Torschützen müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können. Material: - Schöningh: Informatik 2 - Einkaufsliste - Textdatei

		zum Beispiel: Abfahrtslauf Bei einem Abfahrtslauf kommen die Skifahrer nacheinander an und werden nach ihrer Zeit in eine Rangliste eingeordnet. Diese Rangliste wird in einer Anzeige ausgegeben. Ankommende Abfahrer müssen an jeder Stelle der Struktur, nicht nur am Ende oder Anfang eingefügt werden können. Materialien: Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.2 - Listen (Download Q1-II.2)
4. Vertiefung - Anwendungen von Listen, Stapeln oder Schlangen in mindestens einem weiteren Kontext		Beispiel: Skispringen Ein Skispringen hat folgenden Ablauf: Nach dem Sprung erhält der Springer eine Punktzahl und wird nach dieser Punktzahl in eine Rangliste eingeordnet. Die besten 30 Springer qualifizieren sich für den zweiten Durchgang. Sie starten in umgekehrter Reihenfolge gegenüber der Platzierung auf der Rangliste. Nach dem Sprung erhält der Springer wiederum eine Punktzahl und wird nach der Gesamtpunktzahl aus beiden Durchgängen in die endgültige Rangliste eingeordnet. *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1-II.3 – Anwendungen für lineare Datenstrukturen (Download Q1-II.3) oder zum Beispiel: Rangierbahnhof Auf einem Güterbahnhof gibt es drei Gleise, die nur zu einer Seite offen sind. Wagons können also von einer Seite auf das Gleis fahren und nur rückwärts wieder hinausfahren. Die Wagons tragen Nummern, wobei die Nummer jedoch erst eingesehen werden kann, wenn der Wagon der vorderste an der offenen Gleisseite ist. (Zwischen den Wagons herumzuturnen, um die anderen Wagonnummern zu lesen, wäre zu gefährlich.) Zunächst stehen alle Wagons unsortiert auf einem Gleis. Ziel ist es, alle Wagons in ein anderes Gleis zu fahren, so dass dort die Nummern der Wagons vom Gleisende aus aufsteigend in richtiger Reihenfolge sind. Zusätzlich zu diesen beiden

		Gleisen gibt es ein Abstellgleis, das zum Rangieren benutzt werden kann. Material: Schöningh: Informatik 2 oder zum Beispiel: Bowlingcenter Ein Bowlingcenter hat sechs Bowlingbahnen. Zu jeder Bahn gehört eine Rinne, in die die gespielten Kugeln zurückbefördert werden. Die Kugeln haben unterschiedliche Gewichte und dazu jeweils eine passende Farbe. Gesucht ist die Farbe, die in einer Rinne am häufigsten vertreten ist. Sollten zwei Farben gleich häufig vorkommen, wird diejenige mit dem höchsten Gewicht ausgegeben. Material: Schöningh: Informatik 2 oder zum Beispiel. „Turing Tower“ Erstellung einer Steuerungssoftware für einen Fahrstuhl, die mit einer Schlange, Stapel oder einer Liste implementiert werden. Material: net-schulbuch.de - Lineare Datenstrukturen

Unterrichtsvorhaben Q1-IV

Thema: Suchen und Sortieren auf linearen Datenstrukturen
Zeitbedarf: ca. 16 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Suchen von Daten in Listen und Arrays a) Lineare Suche in Listen und in Arrays b) Binäre Suche in Arrays als Beispiel für rekursives Problemlösen c) Untersuchung der beiden Suchverfahren hinsichtlich ihrer Effizienz (Laufzeitverhalten, Speicherbedarf)	Die Schülerinnen und Schüler • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • beurteilen die Effizienz von Algorithmen unter	PAP-Designer Beispiel: Bundesjugendspiele Die Teilnehmer an Bundesjugendspielen nehmen an drei Disziplinen teil und erreichen dort Punktzahlen. Diese werden in einer Wettkampfkarte eingetragen und an das Wettkampfbüro gegeben. Zur Vereinfachung sollte sich das Modell auf die drei Disziplinen „Lauf“, „Sprung“ und „Wurf“ beschränken.

	Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), • entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teilen und Herrschen“ (M), • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), • nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • testen Programme systematisch anhand von Beispielen (I), • stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D).	Im Wettkampfbüro wird das Ergebnis erstellt. Das Programm soll dafür zunächst den Besten einer Disziplin heraussuchen können und später das gesamte Ergebnis nach gewissen Kriterien sortieren können. oder Beispiel: Musikerverwaltungsprogramm Materialien: - Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren - Suchen – iterativ und rekursiv Schöningh: Informatik 2
2. Sortieren in Listen und Arrays - Entwicklung und Implementierung von iterativen und rekursiven Sortierverfahren a) Entwicklung und Implementierung eines einfachen Sortierverfahrens für eine Liste b) Implementierung eines einfachen Sortierverfahrens für ein Feld c) Entwicklung eines rekursiven Sortierverfahren für ein Feld (z.B. Sortieren durch Mischen)		Beispiel: Zahlenliste Beispiel: Informatikerliste Materialien: - net-schulbuch.de: Sortierverfahren - Schöningh: Informatik 1/2: Sortieren – iterativ und rekursiv Beispiel: Karteiverwaltung *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q1.3 - Suchen und Sortieren (Download Q1-III.1)
3. Untersuchung der Effizienz der Sortierverfahren „Sortieren durch direktes Einfügen“ und „Quicksort“ auf linearen Listen a) Grafische Veranschaulichung der Sortierverfahren b) Untersuchung der Anzahl der Vergleichsoperationen und des Speicherbedarfes bei beiden Sortierverfahren *c)* Beurteilung der Effizienz der beiden Sortierverfahren		Beispiel: Zahlenliste Beispiel: Informatikerliste Materialien: - net-schulbuch.de: Sortierverfahren - Schöningh: Informatik 2: Sortieren – iterativ und rekursiv

Unterrichtsvorhaben Q1-V

Thema: Modellierung und Nutzung von relationalen Datenbanken in Anwendungskontexten

Zeitbedarf: ca. 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanken und Grundbegriffe • Entwicklung von Fragestellungen zur vorhandenen Datenbank • Analyse der Struktur der vorgegebenen Datenbank und Erarbeitung der Begriffe Tabelle, Attribut, Datensatz, Datentyp, Primärschlüssel, Fremdschlüssel, Datenbankschema b) SQL-Abfragen • Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) …FROM, WHERE, AND, OR, NOT) auf einer Tabelle • Analyse und Erarbeitung von SQL-Abfragen auf einer und mehrerer Tabelle zur Beantwortung der Fragestellungen (JOIN, UNION, AS, GROUP BY,ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (…), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL) c) Vertiefung an einem weiteren Datenbankbeispiel	Die Schülerinnen und Schüler • erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), • analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), • analysieren und erläutern eine Datenbankmodellierung (A), • erläutern die Eigenschaften normalisierter Datenbankschemata (A), • bestimmen Primär- und Sekundärschlüssel (M), • ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), • modifizieren eine Datenbankmodellierung (M), • modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), • bestimmen Primär- und Sekundärschlüssel (M), • überführen Datenbankschemata in vorgegebene Normalformen (M), • verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einen 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),	Beispiel: Schulverwaltung In einer Software werden Kurse, Klassen, Schüler, Lehrer und Noten einer Schule verwaltet. Man kann dann ablesen, welcher Schüler von welchem Lehrer unterrichtet wurde und welche Note er ihm gegeben hat. Dazu ist die Datenbank zu modellieren, ggf. zu normalisieren und im Datenbanksystem umzusetzen. Weiter sollen sinnvolle Abfragen entwickelt werden und das Thema Datenschutz besprochen werden. oder Beispiel: Bibliotheksverwaltung Materialien: - net-schulbuch.de: Datenbanken - Schöningh: Informatik 2: Datenbanken
2. Modellierung von relationalen Datenbanken a) Entity-Relationship-Diagramm • Ermittlung von Entitäten, zugehörigen Attributen, Relationen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms • Erläuterung und Modifizierung einer Datenbankmodellierung		Beispiel: Schulverwaltung s.o. zum Beispiel: Lottodatenbank zum Beispiel: Mietshäuserverwaltung zum Beispiel: Schreibwarenladen Die Datenverwaltung eines Schreibwarenladens soll in einem Datenbanksystem umgesetzt werden. Ausgehend von der Modellierung soll mit Hilfe eines

b) Entwicklung einer Datenbank aus einem Datenbankentwurf • Modellierung eines relationalen Datenbankschematas zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln c) Redundanz, Konsistenz und Normalformen • Untersuchung einer Datenbank hinsichtlich Konsistenz und Redundanz in einer Anwendungssituation • Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)	• überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	ER-Modells und eines Datenbankschemas dieser erste Entwurf normalisiert und in einem Datenbanksystem umgesetzt werden. Es schließen sich diverse SQL-Abfragen an, wobei auf die Relationenalgebra eingegangen wird. Projekt: Test erstellen und durchführen Datenbankanbindung an Java Materialien: - net-schulbuch.de: Datenbanken - Schöningh: Informatik 2: Datenbanken

Unterrichtsvorhaben Q1-V

Thema: Sicherheit und Datenschutz in Netzstrukturen
Zeitbedarf: ca. 10 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien 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 asymmetrische kryptografische Verfahren (Cäsar-, Vigenère-, RSA-Verfahren) als Methoden Daten im Netz verschlüsselt zu übertragen	Die Schülerinnen und Schüler • beschreiben und erläutern Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), • analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A), • untersuchen und bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen, die Sicherheit von Informatik-systemen sowie die Einhaltung der Datenschutzbestimmungen und des Urheberrechts (A), • untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung	Cäsar-Verfahren Vigenere-Verfahren Häufigkeitsanalyse Material: Matheprisma: - Cäsar-Chiffren - Modul RSA *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben, Verschlüsselung Q1.5 - Zugriff auf Daten in Netzwerken (Download Q1-V.1) - FILIUS

2. Fallbeispiele zur Datenschutzproblematik und zum Urheberrecht	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

Unterrichtsvorhaben Q2-I

Thema: Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen
Zeitbedarf: ca. 24 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. **Analyse von Baumstrukturen in verschiedenen Kontexten** a) Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) b) Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten	Die Schülerinnen und Schüler • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), • analysieren und erläutern Algorithmen und Programme (A), • beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), • 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),	Beispiel: Morsebaum Das Morsealphabet wird anhand des Kriteriums Punkt (links) und Strich (rechts) in einem Binärbaum codiert. Mit Hilfe dieses Morsebaumes sollen Morsenachrichten decodiert und codiert werden. oder Beispiel: Termbaum Der Aufbau von Termen wird mit Hilfe von binären Baum-strukturen verdeutlicht. oder Beispiel: Ahnenbaum Die binäre Baumstruktur ergibt sich daraus, dass jede Person genau einen Vater und eine Mutter hat. Weitere Beispiele für Anwendungskontexte für binäre Bäume: Beispiel: Suchbäume (zur sortierten Speicherung von Daten) Alle Inhalte, die nach einer Ordnung vor dem Inhalt im aktuellen Teilbaum stehen, sind in dessen linkem Teilbaum, alle die nach dem Inhalt im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) oder Beispiel: Entscheidungsbäume Um eine Entscheidung zu treffen, werden mehrere Fragen mit ja oder nein beantwortet. Die Fragen, die möglich sind, wenn die Antwort auf eine Frage mit „ja“ beantwortet wird, befinden sich im linken Teil-baum,

	• 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),	die Fragen, die möglich sind, wenn die Antwort „nein“ lautet, stehen im rechten Teilbaum. oder Beispiel: Codierungsbäume für Codierungen, deren Alphabet aus genau zwei Zeichen besteht Morse hat Buchstaben als Folge von Punkten und Strichen codiert. Diese Codierungen können in einem Binärbaum dargestellt werden, so dass ein Übergang zum linken Teilbaum einem Punkt und ein Übergang zum rechten Teilbaum einem Strich entspricht. Wenn man im Gesamtbaum startet und durch Übergänge zu linken oder rechten Teilbäumen einen Pfad zum gewünschten Buchstaben sucht, erhält man die Morsecodierung des Buchstabens. Materialien: - net-schulbuch.de: Baumstrukturen - Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum - Schöningh: Informatik 2: Nicht-Lineare Datenstrukturen
2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms c) Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen d) Implementierung der Anwendung oder von Teilen der Anwendung e) Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf		Beispiel: Informatiker- bzw. Personenbaum als binärer Baum In einem binären Baum werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. Alle Namen, die nach dieser Ordnung vor dem Namen im aktuellen Teilbaum stehen, sind in dessen linkem Teil-baum, alle die nach dem Namen im aktuellen Teilbaum stehen, sind in dessen rechtem Teilbaum. (Dies gilt für alle Teilbäume.) Folgende Funktionalitäten werden benötigt: • Einfügen der Informatiker-Daten in den Baum • Suchen nach einem Informatiker über den Schlüssel Name • Ausgabe des kompletten Datenbestands in nach Namen sortierter Reihenfolge Materialien: - net-schulbuch.de: Baumstrukturen - Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärbaum

3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree a) Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen b) Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, c) grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften d) Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation e) Implementierung der Anwendung oder von Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums		Mögliches Beispiel: Suchbaum mit Ganzzahlen Mögliches Beispiel: Informatikerbaum als Suchbaum In einem binären Suchbaum werden die Namen und die Geburtsdaten von Informatikern lexikographisch geordnet abgespeichert. *Materialien:* - net-schulbuch.de: Baumstrukturen - Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.1 – Binärer Suchbaum
4. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen		Mögliches Beispiel: Codierungsbäume oder Huffman-Codierung mögliches Beispiel: Buchindex oder eines der zuvor nicht behandelten Bäume (s.o.)

Unterrichtsvorhaben Q2-II

Thema: Endliche Automaten und formale Sprachen

Zeitbedarf: ca. 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien Materialien
1. Endliche Automaten *a)* Vom Automaten in den Schülerinnen und Schülern bekannten Kontexten zur formalen Beschreibung eines endlichen Automaten b) Untersuchung, Darstellung und Entwicklung endlicher Automaten	Die Schülerinnen und Schüler • analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens auf bestimmte Eingaben (A), • analysieren und erläutern Grammatiken regulärer Sprachen (A), • zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A), • ermitteln die formale Sprache, die durch eine Grammatik erzeugt wird (A), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M), • entwickeln zur akzeptierten Sprache eines Automaten die zugehörige Grammatik (M), • entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M), • modifizieren Grammatiken regulärer Sprachen (M), • entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M), • stellen endliche Automaten in Tabellen oder Graphen dar und überführen sie in die jeweils andere Darstellungsform (D), • ermitteln die Sprache, die ein endlicher Automat akzeptiert (D). • beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D).	https://flaci.com *Beispiele:* Cola-Automat, Akzeptor für bestimmte Zahlen, Akzeptor für Teilwörter in längeren Zeichenketten, Akzeptor für Terme Materialien: - net-schulbuch.de: Automaten - Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.2 – Endliche Automaten, Formale Sprachen
2. Untersuchung und Entwicklung von Grammatiken regulärer Sprachen a) Erarbeitung der formalen Darstellung regulärer Grammatiken b) Untersuchung, Modifikation und Entwicklung von Grammatiken c) Entwicklung von endlichen Automaten zum Erkennen regulärer Sprachen die durch Grammatiken gegeben werden d) Entwicklung regulärer Grammatiken zu endlichen Automaten		https://flaci.com Beispiele: reguläre Grammatik für Wörter mit ungerader Parität, Grammatiken für Bitstrings, die als Dualzahl interpretiert durch zwei oder durch drei teilbar sind, Grammatik für die Sprache aller vollständig geklammerten arithmetischen Ausdrücke Materialien: (s.o.)
3. Grenzen endlicher Automaten		Beispiele: Klammerausdrücke, $a^n b^n$ im Vergleich zu $(ab)^n$

Unterrichtsvorhaben Q2-III

Thema: Prinzipielle Arbeitsweise eines Computers und Grenzen der Automatisierbarkeit

Zeitbedarf: ca. 12 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien 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 Informatikystemen (A).	*Beispiel:* Addition von 4 zu einer eingegeben Zahl mit einem Rechnermodell *Materialien:* Ergänzungsmaterialien zum Lehrplannavigator Unterrichtsvorhaben Q2.3 –Von-Neumann-Architektur und maschinennahe Simulationssoftware „Johnny“ Beispiele: Addition zweier Zahlen, Multiplikation durch Addition
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

Unterrichtsvorhaben Q2-III

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

2.2 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

2.2.1 Vorbemerkungen

Die rechtlich verbindlichen Vorgaben zur Leistungsbewertung finden sich in § 48 SchulG, in § 6 der APO-SI sowie den §§ 13-17 der APO-GOSt. Danach ist eine rein rechnerische Bildung von Abschlussnoten unzulässig. Die Fachkonferenz legt nach § 70 SchulG Grundsätze zu Verfahren und Kriterien der Leistungsbewertung fest. Sie orientiert sich dabei an den in den Lehrplänen für die Sekundarstufen I und II ausgewiesenen Kompetenzen.

Das fachbezogene Leistungskonzept ist für alle Mitglieder einer Fachschaft verbindlich. Es soll für ein möglichst hohes Maß an Transparenz und Vergleichbarkeit von Leistungsbeurteilungen sorgen. Rückfragen zum Leistungsstand richten Schülerinnen und Schüler sowie Eltern bitte immer zunächst an die unterrichtenden Fachlehrerinnen und Fachlehrer.

Die Grundsätze der Leistungsbewertung werden den Schülerinnen und Schülern immer zum Schuljahresbeginn, bei Lehrerwechsel auch zum Halbjahresbeginn mitgeteilt. Ein Hinweis darauf wird im Klassenbuch / Kursheft vermerkt.

Kriterien der Leistungsbewertung im Zusammenhang mit konkreten, insbesondere offenen Arbeitsformen werden den Schülerinnen und Schülern grundsätzlich vor deren Beginn transparent gemacht.

Jede Lehrerin / jeder Lehrer dokumentiert regelmäßig und kontinuierlich die von den Schülerinnen und Schülern erbrachten Leistungen.

Die Leistungsrückmeldung erfolgt in regelmäßigen Abständen (zumindest zum Quartalsende) differenziert und individuell in schriftlicher oder mündlicher Form.

Bei Minderleistungen erhalten die Schülerinnen und Schüler sowie ihre Erziehungsberechtigten im Zusammenhang mit den Halbjahreszeugnissen individuelle Lern- und Förderempfehlungen, die die Lernenden - ihrem jeweiligen Leistungsstand entsprechend - zum Weiterlernen ermutigen, indem sie Hinweise zu Erfolg versprechenden individuellen Lernstrategien geben. Den Eltern werden im Rahmen der Lern– und Förderempfehlungen Wege aufgezeigt, wie sie das Lernen ihrer Kinder unterstützen können.

Bei Elternsprechtagen und im Rahmen von Sprechstunden erhalten die Erziehungsberechtigten Gelegenheit, sich über den Leistungsstand ihrer Kinder zu informieren und dabei Perspektiven für die weitere Lernentwicklung zu besprechen.

2.2.2 Bewertungsbereich Sonstige Mitarbeit

Die Beurteilung der sonstigen Mitarbeit erfolgt gemäß dem Lehrplan Informatik. Sie erfasst die Qualität, die Quantität und die Kontinuität verschiedener Beiträge. Für die Bewertung der Leistungen sind sowohl Inhalts- als auch Darstellungsleistungen zu berücksichtigen.

Die sonstige Mitarbeit wird dabei in einem kontinuierlichen Prozess während des Schulhalbjahres festgestellt.

Formen der sonstigen Mitarbeit

- Mündliche Beteiligung am Unterrichtsgespräch

- Regelmäßige Beteiligung (quantitativ und qualitativ)
- Wiedergabe von Kenntnissen und Prozessen
- Darstellung von Problemsituationen
- Beiträge zur Entwicklung von Problemlösungen und Bewertung von Arbeitsständen und –Ergebnissen
- Vorstellung von Arbeitsergebnissen aus Gruppen- oder Partnerarbeit
- Nutzung adäquater Fachsprache

- Selbstständige Arbeit, zu zweit oder in Gruppen.

- Mitarbeit bei der Organisation (Absprachen, Arbeitseinteilung, Schnittstellenvereinbarung, gegenseitige Hilfen) der gemeinsamen Arbeit
- Kooperatives Verhalten und gemeinsame konstruktive Suche nach Lösungswegen bei den gestellten Problemen
- Erreichen des Arbeitszieles unter Einsatz der zur Verfügung stehenden Materialien und der erlernten Arbeitstechniken

- Praktische Arbeit am Computer

- Vor- und Nachbereitung der Rechnerarbeitsphasen (u.a. Dokumentationen)
- Kooperatives Arbeiten am Rechner und Beratung anderer Arbeitsgruppen

- Sinnstiftender Umgang mit den vorhandenen Ressourcen (Nutzung von Hardware, Werkzeugen, insbesondere Umgang mit der Programmierumgebung, angemessene Reaktion auf Fehlermeldungen usw.)

- Anfertigung und Präsentation von Hausaufgaben

- Vor- und Nachbereitung von Unterrichtsstunden
- Intensive und möglichst selbstständige Auseinandersetzung mit Materialien und schriftlichen Informationen
- Selbstständige Weiterarbeit und Vervollständigung von Programmierlösungen (auch mit Hilfe des SLZ).

- Bis zu einer schriftlichen Übung pro Halbjahr

- Evtl. Erstellen von Protokollen

- Evtl. Referate

Die Reihenfolge der oben angegebenen Punkte spiegelt die Bedeutung bei der Beurteilung im Sinne einer Rangfolge wider, wobei nach konkretem Unterrichtsverlauf die Gewichtungen abweichen können.

2.2.3 Klausuren

Es ist Beschluss der Fachschaft, im 1. Halbjahr der Einführungsphase nur eine Klausur (2. Quartal) zu schreiben. Die Dauer der Klausuren ist laut APO-GOSt vorgegeben.
Kompetenzen: Analyse, Algorithmik, Programmiertechnik, Modellierung, Umgang mit den vorhandenen Ressourcen
Materialgrundlage: vorhandene Ressourcen
Aufgabenformate: Reproduktive Aufgabenteile, Modellierung und Realisierung von Programmen
zugelassene Hilfsmittel: keine

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

- Grundkurse Q 2.2: 1 Klausur unter Abiturbedingungen
- Anstelle einer Klausur kann gemäß dem Beschluss der Lehrerkonferenz in Q 1.2 eine Facharbeit geschrieben werden.

Übersicht über Gewichtung/Punkteverteilung - Bewertungskriterien

Jede Aufgabe wird nach Bedeutung der überprüften Lernziele, Anteil der erwarteten Arbeitszeit, Umfang und Komplexität der einzubringenden Teilleistungen, Grad der geforderten Selbstständigkeit und Art und Form der Darstellung entsprechend bepunktet und dieses den Schülerinnen und Schülern offengelegt. Der Hilfscharakter eines solchen Bewertungsverfahrens ist zu betonen und die Gesamtnote ist nicht unbedingt und unflexibel aus der erreichten Prozentzahl zu bestimmen.

Notentabelle:
Durch Leistungen mit vorwiegend wiederholendem Charakter ist eine ausreichende Bewertung zu erzielen. In der Regel reichen 40% der Maximalpunktzahl zu einem ausreichenden Ergebnis. Die Notenstufen ergeben sich i.a. daraus aus äquidistanter Einteilung.

Die Dokumentation der Leistungsbewertung enthält Kriterienraster.

2.2.4 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

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

Überfachliche Grundsätze:

1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
4) Medien und Arbeitsmittel sind schülernah gewählt.

5) Die Schüler/innen erreichen einen Lernzuwachs.
6) Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
7) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
8) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
9) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
10) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
11) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
12) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
13) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
14) Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

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

3 Qualitätssicherung und Evaluation

Aufgrund der „kleinen“ Fachschaft und der z.T. neu eingeführten Unterrichtsmaterialien dem Schulbuch Informatik aus dem Schöningh-Verlag und dem „net-schulbuch.de“ werden weitere Erfahrungen gesammelt, um zu ermitteln welche Unterrichtsmaterialien dauerhaft im Unterricht eingesetzt werden. Diese werden ständig neu bewertet und ggf. ersetzt. Auch eine Evaluation durch die Schülerinnen und Schüler wird dazu durchgeführt.