

Gymnasium Lohmar



Schulinterner Lehrplan

Informatik

Stand: September 2016

Inhaltsverzeichnis

1 Die Fachgruppe Informatik des Gymnasiums Lohmar.....	3
2 Entscheidungen zum Unterricht.....	4
2.1 Unterrichtsvorhaben.....	4
2.1.1 Übersichtsraster Unterrichtsvorhaben.....	5
2.1.1.1 Einführungsphase – Grundkurs.....	5
2.1.1.2 Qualifikationsphase 1 – Grundkurs.....	7
2.1.1.3 Qualifikationsphase 2 – Grundkurs.....	8
2.1.2 Konkretisierte Unterrichtsvorhaben.....	9
2.1.2.1 Einführungsphase - Unterrichtsvorhaben Nr. 1.....	10
2.1.2.2 Einführungsphase - Unterrichtsvorhaben Nr. 2.....	11
2.1.2.3 Einführungsphase - Unterrichtsvorhaben Nr. 3.....	12
2.1.2.4 Einführungsphase - Unterrichtsvorhaben Nr. 4.....	14
2.1.2.5 Qualifikationsphase I - Unterrichtsvorhaben Nr. 1.....	16
2.1.2.6 Qualifikationsphase I - Unterrichtsvorhaben Nr. 2.....	18
2.1.2.7 Qualifikationsphase I - Unterrichtsvorhaben Nr. 3.....	21
2.1.2.8 Qualifikationsphase I - Unterrichtsvorhaben Nr. 4.....	23
2.1.2.9 Qualifikationsphase II - Unterrichtsvorhaben Nr. 1.....	25
2.1.2.10 Qualifikationsphase II - Unterrichtsvorhaben Nr. 2.....	28
2.1.2.11 Qualifikationsphase II - Unterrichtsvorhaben Nr. 3.....	31
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	33
2.3 Grundsätze der Leistungsbewertung.....	34
2.3.1 Transparenz der Leistungsbeurteilung	34
2.3.2 Grundsätze der Leistungsbeurteilung	34
2.3.3 Formen der Leistungsüberprüfung.....	35
2.3.3.1 Kursarbeiten bzw. Klausuren	35
2.3.3.2 Mitarbeit im Unterricht	36
2.3.3.3 Individuelle Förderung	36
2.3.3.4 Bildung der Zeugnisnote	36
2.4 Lehr- und Lernmittel.....	37
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen.....	37
4 4 Qualitätssicherung und Evaluation.....	37

1 Die Fachgruppe Informatik des Gymnasiums Lohmar

Das Fach Informatik wird am Gymnasium Lohmar ab der Jahrgangsstufe 8 im Wahlpflichtbereich II unterrichtet. In diesem zweijährigen Kurs werden in altersstufengerechter Weise unter anderem die Grundlagen der Algorithmik am Beispiel verschiedener didaktischer Lernumgebungen sowie die technische Informatik am Beispiel von Schaltwerken und Schaltnetzen thematisiert. Darüber hinaus erhalten die Schülerinnen und Schüler dieses Differenzierungskurses einen tiefergehenden Einblick in Tabellenkalkulationen sowie die Gestaltung und Programmierung von Internetseiten.

Das Fach Informatik wird in der Sekundarstufe II am Gymnasium Lohmar als Wahlfach angeboten. In der Einführungsphase und Qualifikationsphase können die Schülerinnen und Schüler das Fach als dreistündigen Grundkurs belegen. Darüber hinaus bietet das Gymnasium Lohmar den Schülerinnen und Schülern, welche den Grundkurs Informatik in der Qualifikationsphase belegt haben, die Möglichkeit, einen Projektkurs zu belegen. Dieser Projektkurs ist so angelegt, dass sich die Schülerinnen und Schüler – bezogen auf ein Rahmenthema – individuellen Vorhaben im Rahmen der „Softwareentwicklung“ widmen können, die sie weitgehend selbstständig planen und an denen sie möglichst eigenständig „projektartig“ arbeiten.

Um insbesondere Schülerinnen und Schülern, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, gerecht zu werden, wird in den Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Bestehen des Kurses erforderlich sind. Im schulinternen Curriculum der Sekundarstufe I wurde darauf geachtet, dass die dort behandelten Inhalte keine unmittelbaren inhaltlichen Vorkenntnisse für die Sekundarstufe II schaffen. Damit soll ein schulinterner Wissensvorsprung (inhaltsbezogen) einzelner Schülerinnen und Schülern, die in der Sekundarstufe I Informatik belegt haben, verhindert werden. Ein Kompetenzvorsprung ist zu erwarten.

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

Darüber hinaus trägt er zu einer breitgefächerten Allgemeinbildung bei, bietet gleichzeitig Raum für individuelle Spezialisierungen und ermöglicht verantwortungsvolles Handeln in einer sich schnell wandelnden und von technischen Fortschritten geprägten Welt.

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 stellt einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik des Gymnasiums Lohmar aus drei Lehrkräften mit Sek-II-Lehrbefähigung. Das Gymnasium verfügt über zwei Computerräume mit jeweils 17 Windows-Computerarbeitsplätzen und einem Selbstlernzentrum mit 9 Arbeitsplätzen, Alle Fachräume sowie Oberstufenräume sind mit Beamer und einem Computerarbeitsplatz ausgestattet. Alle Computerarbeitsplätze schulinternen Rechnernetz angeschlossen. Alle Schülerinnen und Schüler besitzen einen individuell Zugang sowie eigenen Datenbereich. Darüberhinaus können alle Schülerinnen und Schüler der Oberstufe das schulische WLAN mit ihren privaten Endgeräten nutzen.

In den Kursen des Faches Informatik werden zwischen 16 und 32 Schülerinnen und Schüler unterrichtet. Projektkurse sind in der Regel kleiner. Die Arbeit am Rechner erfolgt also im Normalfall in Einzel- oder Partnerarbeit.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

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

Im "Übersichtsraster Unterrichtsvorhaben" (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnungen 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 Spielraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans nur 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, besitzt die exemplarische Ausweisung "konkretisierter Unterrichtsvorhaben" (Kapitel 2.1.2) empfehlenden Charakter. 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, fachübergreifenden Kooperationen, Lernmitteln und Lernorten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind. Abweichungen von den vorgeschlagenen Vorgehensweisen bezüglich der konkretisierten Unterrichtsvorhaben sind im Rahmen der pädagogischen Freiheit der Lehrkräfte jederzeit möglich. Sicherzustellen bleibt allerdings auch hier, dass im Rahmen der Umsetzung der Unterrichtsvorhaben insgesamt alle konkretisierten Kompetenzerwartungen des Kernlehrplans Berücksichtigung finden.

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

2.1.1 Übersichtsraster Unterrichtsvorhaben

2.1.1.1 Einführungsphase – Grundkurs

Im Folgenden sollen Unterrichtsvorhaben für das Fach Informatik dargestellt werden. Alle hier aufgeführten Vorhaben beziehen sich auf Grundkurse in der Einführungsphase.

Zu jedem Unterrichtsvorhaben ist eine Anknüpfung an den Kernlehrplan Informatik in Form von Kompetenzbezügen gegeben. Die aufgeführten Kompetenzen sind dabei so zu verstehen, dass das entsprechende Unterrichtsvorhaben zum Erwerb derselben beiträgt. Kompetenzerwerb ist ein kontinuierlicher und kumulativer Prozess, der sich über längere Zeiträume hinzieht und die wiederholte Beschäftigung mit entsprechenden fachlichen Gegenständen und Themen in variierenden Anwendungssituationen oder auf zunehmenden Anforderungsniveaus voraussetzt. Es kann daher nicht der Anspruch erhoben werden, dass die aufgeführten Kompetenzen nach Abschluss lediglich eines Unterrichtsvorhabens vollständig erworben wurden.

Übersichtsraster Unterrichtsvorhaben: Einführungsphase – Grundkurs:

Unterrichtsvorhaben E-I	Unterrichtsvorhaben E-II
Thema: Überblick über das Fach Informatik, Grundlagen von Informatiksystem und Geschichte der Informatik	Thema: Grundlagen der Programmierung mit Java und einfache Algorithmik
Zentrale Kompetenzen: Argumentieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren	Zentrale Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren
Inhaltsfelder: Informatiksysteme Algorithmen Informatik, Mensch und Gesellschaft	Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatiksysteme
Inhaltliche Schwerpunkte: Digitalisierung, Binärzahlen Internet, Netzwerke Einsatz von Informatiksystemen Einblick in die Graphentheorie	Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache
Zeitbedarf: 6 Stunden	Zeitbedarf: 33 Stunden

Unterrichtsvorhaben E-III	Unterrichtsvorhaben E-IV
<p>Thema: Implementierung einfacher Algorithmen und GUI-Entwurf</p>	<p>Thema: Objektorientierte Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen, Simulationen und grafischen Oberflächen</p>
<p>Zentrale Kompetenzen: Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren</p>	<p>Zentrale Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren</p>
<p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten Informatik, Mensch und Gesellschaft</p>	<p>Inhaltsfelder: Daten und ihre Strukturierung Algorithmen Formale Sprachen und Automaten</p>
<p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p>	<p>Inhaltliche Schwerpunkte: Objekte und Klassen Analyse, Entwurf und Implementierung einfacher Algorithmen Syntax und Semantik einer Programmiersprache</p>
<p>Zeitbedarf: 20 Stunden</p>	<p>Zeitbedarf: 36 Stunden</p>

2.1.1.2 Qualifikationsphase 1 – Grundkurs

Übersichtsraster Unterrichtsvorhaben: Qualifikationsphase 1 – Grundkurs:

Unterrichtsvorhaben Q1-I	Unterrichtsvorhaben Q1-II
Thema: Wiederholung und Vertiefung der objektorientierten Modellierung	Thema: Organisation und Verarbeitung von Daten I - Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen
Zentrale Kompetenzen: Modellieren, Darstellen und Interpretieren, Implementieren, Kommunizieren und Kooperieren	Zentrale Kompetenzen: Modellieren, Implementieren, Darstellen und Interpretieren, Argumentieren
Inhaltsfelder: Daten und ihre Strukturierung, Algorithmen, Informatik, Mensch und Gesellschaft	Inhaltsfelder: Daten und ihre Strukturierung, Algorithmen
Inhaltliche Schwerpunkte: Objekte und Klassen, Wirkung der Automatisierung	Inhaltliche Schwerpunkte: Objekte und Klassen, Syntax und Semantik einer Programmiersprache, Analyse, Entwurf und Implementierung von Algorithmen, Algorithmen in ausgewählten informatischen Kontexten
Zeitbedarf: 14 Stunden	Zeitbedarf: 20 Stunden

Unterrichtsvorhaben Q1-III	Unterrichtsvorhaben Q1-IV
Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen	Thema: Automaten und formale Sprachen
Zentrale Kompetenzen: Argumentieren, Darstellen und Interpretieren, Modellieren, Implementieren	Zentrale Kompetenzen: Argumentieren, Darstellen und Interpretieren, Modellieren, Kommunizieren und Kooperieren
Inhaltsfelder: Algorithmen, Formale Sprachen und Automaten	Inhaltsfelder: Formale Sprachen und Automaten, Informatiksysteme
Inhaltliche Schwerpunkte: Analyse, Entwurf und Implementierung von Algorithmen. Algorithmen in ausgewählten informatischen Kontexten, Syntax und Semantik einer Programmiersprache	Inhaltliche Schwerpunkte: Syntax und Semantik einer Programmiersprache, Endliche Automaten, Grammatiken regulärer Sprachen, Möglichkeiten und Grenzen von Automaten und formalen Sprachen
Zeitbedarf: 20 Stunden	Zeitbedarf: 20 Stunden

2.1.1.3 Qualifikationsphase 2 – Grundkurs

Übersichtsraster Unterrichtsvorhaben: Qualifikationsphase 2 – Grundkurs:

Unterrichtsvorhaben Q2-I	Unterrichtsvorhaben Q2-II
Thema: Modellierung und Implementierung dynamische nichtlineare Datenstrukturen am Beispiel der Binärbäume	Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten
Zentrale Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren, <i>Kommunizieren und Kooperieren</i>	Zentrale Kompetenzen: Argumentieren, Modellieren, Implementieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren
Inhaltsfelder: Daten und ihre Strukturierung, Algorithmen, Formale Sprachen und Automaten, Informatiksysteme	Inhaltsfelder: Daten und ihre Strukturierung, Algorithmen, Formale Sprache und Automaten, Informatiksysteme, Informatik, Mensch und Gesellschaft
Inhaltliche Schwerpunkte: Objekte und Klassen, Analyse, Entwurf und Implementierung von Algorithmen, Algorithmen in ausgewählten informatischen Kontexten, Syntax und Semantik einer Programmiersprache, Nutzung von Informatiksystemen	Inhaltliche Schwerpunkte: Datenbanken, Algorithmen in ausgewählten informatischen Kontexten, Syntax und Semantik einer Programmiersprache, Nutzung von Informatiksystemen, Sicherheit, Wirkung der Automatisierung
Zeitbedarf: 15 Stunden	Zeitbedarf: 25 Stunden

Unterrichtsvorhaben Q2-III	
Thema: Sicherheit und Datenschutz in Netzstrukturen	
Zentrale Kompetenzen: Argumentieren, Darstellen und Interpretieren, Kommunizieren und Kooperieren	
Inhaltsfelder: Informatiksysteme, Informatik, Mensch und Gesellschaft	
Inhaltliche Schwerpunkte: Einzelrechner und Rechnernetzwerke, Sicherheit, Nutzung von Informatiksystemen, Wirkungen der Automatisierung	
Zeitbedarf: 15 Stunden	

2.1.2 Konkretisierte Unterrichtsvorhaben

Hinweis:

Thema, Inhaltsfelder, inhaltliche Schwerpunkte, Kompetenzen und Absprachen zur vorhabenbezogenen Konkretisierung hat die Fachkonferenz verbindlich vereinbart. In allen anderen Bereichen (Unterrichtssequenzen und verwendeten Beispiele, Medien und Materialien) sind Abweichungen von den vorgeschlagenen Vorgehensweisen möglich. Darüber hinaus enthält dieser schulinterne Lehrplan in den Kapiteln 2.2 bis 2.3 übergreifende sowie z.T. auch jahrgangsbezogene Absprachen zur fachmethodischen und fachdidaktischen Arbeit, zur Leistungsbewertung und zur Leistungsrückmeldung. Je nach internem Steuerungsbedarf können solche Absprachen auch vorhabenbezogen vorgenommen werden.

Im Folgenden sollen die in Abschnitt 2.1 aufgeführten Unterrichtsvorhaben konkretisiert werden. Diese Konkretisierung hat vorschlagenden Charakter, ohne die pädagogische Freiheit des Lehrenden einschränken zu wollen.

Die übergeordneten Kompetenzen des Kompetenzbereichs "Kommunizieren und Kooperieren" werden in jedem Unterrichtsvorhaben erworben bzw. vertieft und sind daher nicht jedes Mal erneut aufgeführt.

Kommunizieren und Kooperieren (K)

Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte,
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit,
- präsentieren Arbeitsabläufe und Arbeitsergebnisse.

2.1.2.1 Einführungsphase - Unterrichtsvorhaben Nr. 1

Thema: Überblick über das Fach Informatik, Grundlagen von Informatiksystem und Geschichte der Informatik

Leitfragen: Mit welchen Themen befasst sich das Fach Informatik in der Schule? Wie funktioniert ein moderner Computer? Welche Entwicklung durchlief die moderne Datenverarbeitung?

Zeitbedarf: 6 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Da einige Schülerinnen und Schüler das Fach zum ersten Mal in der Einführungsphase belegen, wird zu Beginn ein Überblick über die Themen des Schulfachs Informatik gegeben. Unter anderem wird auf den zentralen Begriff der Information eingegangen und die Möglichkeit der Codierung von Daten, insbesondere wird die Binärdarstellung von Zahlen thematisiert. Stationen der geschichtlichen Entwicklung werden angesprochen wie z.B. prinzipieller Prozessoraufbau, von-Neumann-Architektur und das EVA-Prinzip. Außerdem werden die SuS in die konkrete Nutzung der Informatiksysteme an der Schule eingewiesen, insbesondere gehört dazu die Nutzung einer Lernplattform.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Allgemeine Einführung a) Übersicht über das Fach b) Einführung in die Informatiksysteme der Schule, insbesondere Lernplattform	Die Schülerinnen und Schüler <ul style="list-style-type: none">• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D),• nutzen das verfügbare Informatiksystem zu strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K),• nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	Lernplattform moodle
2. Grundlagen von Informatiksystemen und Überblick über die Geschichte der Informatik a) Darstellung von Zahlen im Binärsystem b) Von-Neumann-Architektur c) Geschichte der Datenverarbeitung	Die Schülerinnen und Schüler <ul style="list-style-type: none">• stellen ganze Zahlen und Zeichen in Binärcodes dar (D),• interpretieren Binärcodes als Zahlen und Zeichen (D),• beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A).• erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A).	

2.1.2.2 Einführungsphase - Unterrichtsvorhaben Nr. 2

Thema: Grundlagen der Programmierung mit Java und einfache Algorithmik

Leitfragen: Aus welchen Bausteinen besteht ein Java-Programm? Welche Datentypen und Kontrollstrukturen stehen in Java zur Verfügung und wie nutzt man diese? Wie verläuft der Entwicklungsprozess eines Java-Programms?

Zeitbedarf: 33 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Anhand des Aachener Leitprogramms „Einführung in die Programmierung mit Java“ werden die Grundlagen zur Programmieretechnik mit Java durch die SuS selbstständig und individuell erarbeitet. Dazu zählen: einfache Datentypen, Variablenkonzept, Wertzuweisungen, Kontrollstrukturen, Methodenkonzept, Arrays. Die Schülerinnen und Schüler wenden ihr Wissen kontextbezogen auf kleinere Problemstellungen an. Dabei durchlaufen sie den kontinuierlichen Prozess von Implementieren, Compilieren und Testen.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Einführung in die Programmierung a) Kap. 1: Erste Schritte b) Kap. 2: Programmaufbau c) Kap. 3: Datentypen und Operationen d) Kap. 4: Variablen e) Kap. 5: Verzweigungen f) Kap. 6: Schleifen g) Kap. 7: Arrays h) Kap. 8: Methoden	Die Schülerinnen und Schüler <ul style="list-style-type: none">• analysieren und erläutern einfache Algorithmen und Programme (A),• modifizieren einfache Algorithmen und Programme (I),• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),• testen Programme schrittweise anhand von Beispielen (I).• implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I).	Leitprogramm „Einführung in die Programmierung“ JavaEditor Lernplattform moodle

2.1.2.3 Einführungsphase - Unterrichtsvorhaben Nr. 3

Thema: Implementierung einfacher Algorithmen und GUI-Entwurf

Leitfragen: Welche Strategien aus dem Erfahrungsschatz der Schülerinnen und Schüler lassen sich algorithmisch beschreiben und programmiertechnisch realisieren? Wie lassen sich benutzerfreundliche Windowsanwendungen aus den Algorithmen entwickeln?

Zeitbedarf: 20 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Verschiedene Algorithmen (z. B. ggT, kgV, Primfaktorzerlegung, Teilermenge, etc.) werden sprachlich formuliert, modifiziert und implementiert. Der Ablauf von Algorithmen wird z.B. mit Variablenbelegungstabellen dargestellt und somit die Korrektheit des Algorithmus verifiziert.

Mithilfe einer geeigneten Entwicklungsumgebung (z.B. JavaEditor, Netbeans, Eclipse, o.a.) werden grafische Oberflächen erstellt und mit den zuvor erstellten Algorithmen verknüpft. Dabei werden die Ereignisbehandlungsroutinen der Oberflächen-Objekte verwendet und das Geheimnisprinzip angewendet. Dabei findet das Model-View-Controller-Konzept Anwendung.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Mathematische Algorithmen</p> <ul style="list-style-type: none"> • Algorithmus zur ggT-Berechnung durch Schleifen mit Anfangs- und Endbedingung • Wiederverwendbarkeit von Methoden • komplexere Algorithmen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen [oder lineare Datensammlungen] zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • stellen den Zustand eines Objekts dar (D), • analysieren und erläutern einfacher Algorithmen und Programme (A) 	<p>Moodle</p>
<p>2. Erstellung von Windowsanwendungen</p> <ul style="list-style-type: none"> • Gui-Entwurf • Model-View-Controller-Konzept • Erste Darstellung von UML-Implementations-Diagrammen 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D). • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A). 	<p>Moodle</p>

2.1.2.4 Einführungsphase - Unterrichtsvorhaben Nr. 4

Thema: Objektorientierte Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen, Simulationen und grafischen Oberflächen

Leitfragen: Was sind Objekte, was sind Klassen? Wie programmiert man objektorientiert? Was bedeutet Vererbung? Wie lassen sich Animationen und Simulationen optischer Gegenstandsbereiche realisieren? Wie lassen sich komplexere Datenflüsse und Beziehungen zwischen Objekten und Klassen realisieren?

Zeitbedarf: 36 Stunden

Absprachen zur vorhabenbezogene Konkretisierung:

Ausgehend von einem vorgegebenen Greenfootprojekt (z. B. crazy kara) wird die Entwicklungsumgebung Greenfoot eingeführt. Diese wird genutzt, um den Schülerinnen und Schüler die Bedeutung von Objekten und Klassen deutlich zu machen. Innerhalb des Projekts modellieren sie eigene Klassen und differenzieren diese in ihren Attributen und Methoden. Neben der Erzeugung von Objekten erlernen sie auch deren Benutzung und Interaktionen untereinander.

Die Klassen des Projektes schrittweise durch Vererbung konkretisiert und deren Funktion implementiert. Dabei werden die bereits erlernten Grunddatentypen und verschiedenen Arten von Methoden in anderem Zusammenhang angewendet und eingeübt. Insbesondere wird das Geheimnisprinzip vertieft.

Erlerntes wird in einer komplett selbst erstellten Animation oder einem komplett selbst erstellten Spiel reorganisiert. Das Projekt wird dabei geeignet dokumentiert.

Nach Abschluss des Projektes wird die objektorientierte Programmierung anhand des Entwurfs und der Implementierung grafischer Oberflächen mit dem Java-Editor kontextbezogen vertieft. Dabei werden die Ereignisbehandlungsroutinen der Oberflächen-Objekte verwendet und das Geheimnisprinzip angewendet.

Unterrichtssequenzen	zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Grundlagen und Einführung in die OO</p> <p>a) Was ist ein Objekt? Was ist eine Klasse?</p> <p>b) Objekte erzeugen</p> <p>c) Darstellung / Verhalten von Objekten</p> <p>d) Methoden und Attribute</p> <p>e) Objektinteraktion</p> <p>f) Vererbung</p> <p>g) Geheimnisprinzip</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • 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), • modellieren Klassen unter Verwendung von Vererbung (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), • stellen den Zustand eines Objekts dar (D), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D), • analysieren und erläutern eine objektorientierte Modellierung (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). 	<p>Moodle</p> <p>Lehrbuch</p>

2.1.2.5 Qualifikationsphase I - Unterrichtsvorhaben Nr. 1

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

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

Zeitbedarf: 14 Stunden

Vorhabenbezogene Konkretisierung:

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

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

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

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

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

2.1.2.6 Qualifikationsphase I - Unterrichtsvorhaben Nr. 2

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

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

Zeitbedarf: 20 Stunden

Vorhabensbezogene 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.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Die Datenstruktur Feld</p> <p>a) Erarbeitung der Anforderungen an eine Datenstruktur</p> <p>b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays</p> <p>c) Modellierung und Implementierung von Anwendungen</p>	<p>Die Schülerinnen und Schüler...</p> <ul style="list-style-type: none"> - 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) - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), 	<p>Moodle Lehrbuch</p>
<p>2. Die Datenstruktur Schlange</p> <p>a) Modellierung und Implementierung der Verknüpfung von Objekten</p> <p>b) Generische Typen, Trennung von Verwaltung und Inhalt dyn. DS.</p> <p>c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden</p> <p>d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen</p> <p>e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p>	<ul style="list-style-type: none"> - 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) 	
<p>3. Die Datenstruktur Stapel</p> <p>a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden</p> <p>b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i></p> <p>c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array (Palindrom)</i>)</p>	<ul style="list-style-type: none"> - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) 	

<p>4. Die Datenstruktur Liste</p> <p>a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>)</p> <p>b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i></p> <p>c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i></p> <p>d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>)</p>		
<p>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</p>		

2.1.2.7 Qualifikationsphase I - Unterrichtsvorhaben Nr. 3

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

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

Zeitbedarf: 20 Stunden

Vorhabenbezogene Konkretisierung:

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Eigenschaften von Algorithmen</p> <p>a) Qualitätseigenschaften von Algorithmen</p> <p>b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“</p> <p>c) Analyse und Entwicklung von rekursiven Algorithmen</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - 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), 	<p>Moodle Lehrbuch</p>
<p>2. Suchen in Listen und Arrays</p> <p>a) Lineare Suche in Listen und Arrays</p> <p>b) Binäre Suche in einem Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - testen Programme systematisch anhand von Beispielen (I). - implementieren und erläutern iterative und rekursive Such- und Sortierverfahren (I), - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A), - beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), 	
<p>3. Sortieren auf Listen und Arrays</p> <p>a) Entwicklung und Implementierung eines iterativen Sortierverfahrens für eine Liste</p> <p>b) Entwicklung und Implementierung eines rekursiven Sortierverfahrens für ein Array</p> <p>c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf</p>	<ul style="list-style-type: none"> - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I), 	

2.1.2.8 Qualifikationsphase I - Unterrichtsvorhaben Nr. 4

Thema: Automaten und formale Sprachen

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

Zeitbedarf: 20 Std.

Vorhabensbezogene Konkretisierung:

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

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

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Endliche Automaten</p> <p>a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen</p> <p>b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus</p> <p>c) Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - 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) 	<p>Moodle Lehrbuch</p>
<p>2. Grammatiken regulärer Sprachen</p> <p>a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften)</p> <p>b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.</p>	<ul style="list-style-type: none"> - 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) - zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf 	
<p>3. Übungen und Vertiefungen</p> <p>Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>		
<p><i>Projekteinstieg</i></p> <p><i>Erarbeitung der formalen Beschreibung und Überprüfung des Verhaltens eines erkennenden Automaten auf bestimmte Eingaben</i></p>		

2.1.2.9 Qualifikationsphase II - Unterrichtsvorhaben Nr. 1

- Thema:** Modellierung und Implementierung von Anwendungen mit dynamischen, nichtlinearen Datenstrukturen am Beispiel der Binärbäume
- Leitfragen:** Wie können Daten im Anwendungskontext mit Hilfe binärer Baumstrukturen verwaltet werden? Wie kann dabei der rekursive Aufbau der Baumstruktur genutzt werden? Welche Vor- und Nachteile haben Suchbäume für die geordnete Verwaltung von Daten?
- Zeitbedarf:** 15 Stunden

Vorhabenbezogene Konkretisierung:

Anhand von Beispielen für Baumstrukturen werden grundlegende Begriffe eingeführt und der rekursive Aufbau binärer Bäume dargestellt.

Anschließend werden für eine Problemstellung in einem der Anwendungskontexte Klassen modelliert und implementiert. Dabei werden die Operationen der Datenstruktur Binärbaum thematisiert und die entsprechende Klasse BinaryTree der Vorgaben für das Zentralabitur NRW verwendet. Klassen und ihre Beziehungen werden in Entwurfs- und Implementationsdiagrammen dargestellt. Die Funktionsweise von Methoden wird anhand grafischer Darstellungen von Binärbäumen erläutert.

Unter anderem sollen die verschiedenen Baumtraversierungen (Pre-, Post- und Inorder) implementiert werden. Unterschiede bezüglich der Möglichkeit, den Baum anhand der Ausgabe der Bauminhalte via Pre-, In- oder Postorder-Traversierung zu rekonstruieren, werden dabei ebenfalls angesprochen, indem die fehlende Umkehrbarkeit der Zuordnung Binärbaum -> Inorder-Ausgabe an einem Beispiel verdeutlicht wird.

Eine Tiefensuche wird verwendet, um einen in der Baumstruktur gespeicherten Inhalt zu suchen.

Zu einer Problemstellung in einem entsprechenden Anwendungskontext werden die Operationen der Datenstruktur Suchbaum thematisiert und unter der Verwendung der Klasse BinarySearchTree der Vorgaben für das Zentralabitur weitere Klassen oder Methoden in diesem Kontext modelliert und implementiert. Die Suchbäume werden wie zuvor auch grafisch dargestellt.

Die Verwendung von binären Bäumen und Suchbäumen wird anhand weiterer Problemstellungen oder anderer Kontexten weiter geübt.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ul style="list-style-type: none"> • Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) • Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten 	<p>1. Analyse von Baumstrukturen in verschiedenen Kontexten</p> <ul style="list-style-type: none"> • Grundlegende Begriffe (Grad, Tiefe, Höhe, Blatt, Inhalt, Teilbaum, Ebene, Vollständigkeit) • Aufbau und Darstellung von binären Bäumen anhand von Baumstrukturen in verschiedenen Kontexten <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), 	<p>Moodle Lehrbuch</p>
<p>2. Die Datenstruktur Binärbaum im Anwendungskontext unter Nutzung der Klasse BinaryTree</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen im Anwendungskontext • Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramms • Erarbeitung der Klasse BinaryTree und beispielhafte Anwendung der Operationen • Implementierung der Anwendung oder von Teilen der Anwendung • Traversierung eines Binärbaums im Pre-, In- und Postorderdurchlauf 	<ul style="list-style-type: none"> • 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), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie/oder lineare und nichtlineare Datensammlungen zu (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), 	
<p>3. Die Datenstruktur binärer Suchbaum im Anwendungskontext unter Verwendung der Klasse BinarySearchTree</p> <ul style="list-style-type: none"> • Analyse der Problemstellung, Ermittlung von Objekten, ihren Eigenschaften und Operationen • Modellierung eines Entwurfsdiagramms und Entwicklung eines Implementationsdiagramm, • grafische Darstellung eines binären Suchbaums und Erarbeitung der Struktureigenschaften • Erarbeitung der Klasse BinarySearchTree und Einführung des Interface Item zur Realisierung einer geeigneten Ordnungsrelation • Implementierung der Anwendung oder von 	<ul style="list-style-type: none"> • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • 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). • erläutern Operationen dynamischer (linearer oder nicht-linearer) Datenstrukturen (A), 	

Teilen der Anwendung inklusive einer sortierten Ausgabe des Baums		
---	--	--

2.1.2.10 Qualifikationsphase II - Unterrichtsvorhaben Nr. 2

Thema: Modellierung und Nutzung relationaler Datenbanken in Anwendungskontexten

Leitfragen: Wie können Fragestellungen mit Hilfe einer Datenbank beantwortet werden? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

Zeitbedarf: 25 Stunden

Vorhabenbezogene Konkretisierung:

Ausgehend von einer konkreten Anwendungssituation entwickeln die Schülerinnen und Schüler Ideen zur Modellierung von Daten und erkennen die Vorzüge von Datenbanksystemen.

In weiteren Anwendungskontexten müssen Datenbanken entwickelt werden, um Daten zu speichern und Informationen für die Beantwortung von möglicherweise auftretenden Fragen zur Verfügung zu stellen. Dafür ermitteln Schülerinnen und Schüler in den Anwendungssituationen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten und stellen diese in Entity-Relationship-Modellen dar. Entity-Relationship-Modelle werden interpretiert und erläutert, modifiziert und in das Relationale Modell überführt.

An einem Beispiel wird verdeutlicht, dass in Datenbanken Redundanzen unerwünscht sind und Konsistenz gewährleistet sein sollte. Die 1. bis 3. Normalform wird als Gütekriterium für Datenbankentwürfe eingeführt. Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Ausgehend von einer vorhandenen Datenbasis, für die eine Datenbank angelegt und mit den Daten gefüllt wird, entwickeln Schülerinnen und Schüler für sie relevante Fragestellungen, die mit dem vorhandenen Datenbestand beantwortet werden sollen. Zur Beantwortung dieser Fragestellungen wird die vorgegebene Datenbank von den Schülerinnen und Schülern analysiert und die notwendigen Grundbegriffe für Datenbanksysteme sowie die erforderlichen SQL-Abfragen werden erarbeitet.

Mit Hilfe von SQL-Anweisungen können anschließend im Kontext relevante Informationen aus der Datenbank extrahiert werden. Die Operationen der Relationenalgebra werden mit SQL-Abfragen simuliert.

Anhand von Fallbeispielen werden Probleme bei der Nutzung von Datenbanksystemen aufgezeigt und im Hinblick auf gesellschaftliche Auswirkungen diskutiert.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
<p>1. Modellierung von relationalen Datenbanken</p> <ul style="list-style-type: none"> 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 Entwicklung einer Datenbank aus einem Datenbankentwurf: Modellierung eines relationalen Datenbankschemata zu einem Entity-Relationship-Diagramm inklusive der Bestimmung von Primär- und Sekundärschlüsseln 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) 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten mit Kardinalitäten in einem Entity-Relationship-Diagramm grafisch dar (D), modifizieren eine Datenbankmodellierung (M), modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), bestimmen Primär- und Sekundärschlüssel (M), analysieren und erläutern eine Datenbankmodellierung (A), erläutern die Eigenschaften normalisierter Datenbankschemata (A), überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D), überführen Datenbankschemata in die 1. bis 3. Normalform (M). 	<p>Moodle Online Kurs Lehrbuch</p>
<p>2. Nutzung von relationalen Datenbanken</p> <ul style="list-style-type: none"> 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 SQL-Abfragen: Analyse vorgegebener SQL-Abfragen und Erarbeitung der Sprachelemente von SQL (SELECT (DISTINCT) ...FROM, WHERE, AND, OR, NOT) auf einer Tabelle 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> modifizieren eine Datenbankmodellierung (M), bestimmen Primär- und Sekundärschlüssel (M), implementieren ein relationales Datenbankschema als Datenbank (I), analysieren und erläutern eine Datenbankmodellierung (A), ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I). erläutern Eigenschaften und Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), 	

<p>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)</p> <ul style="list-style-type: none"> • Vertiefung an einem weiteren Datenbankbeispiel: Vertiefungen am Beispiel der Relationenalgebra 		
<p>3. Gesellschaftliche Auswirkungen der Nutzung von Datenbanksystemen</p>	<ul style="list-style-type: none"> • 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). 	

2.1.2.11 Qualifikationsphase II - Unterrichtsvorhaben Nr. 3

Thema: Modellierung und Implementierung von Client-Server-Anwendungen

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?

Zeitbedarf: 15 Stunden

Vorhabenbezogene Konkretisierung:

Ausgehend von einer einfachen Echo-Anwendung werden die beteiligten Komponenten (Echo-Server und Echo-Client) entwickelt und unter Verwendung der ZK-Klassen implementiert.

Die Echo-Anwendung wird zu einer Chat-Anwendung erweitert, notwendige Protokolle werden entwickelt und systematisch dargestellt.

Die Schülerinnen und Schüler entwickeln eine individuelle Client-Server-Anwendung, definieren notwendige Protokolle und erweitern die Chat-Anwendung entsprechend der Vorgaben.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Entwicklung einer Echo-Anwendung <ul style="list-style-type: none"> • Grundlegende Begriffe • ZA-Klassen 	Die Schülerinnen und Schüler <ul style="list-style-type: none"> • ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), • modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), • verwenden bei der Modellierung geeigneter Problemstellungen Möglichkeiten der Polymorphie (M), • ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), • stellen die Kommunikation zwischen Objekten grafisch dar (D), • stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen (D), • analysieren und erläutern objektorientierte Modellierungen (A), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I). • modifizieren Algorithmen und Programme (I), • implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), • testen Programme systematisch anhand von Beispielen (I). • beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), 	Moodle Lehrbuch Ein Terminal (Putty)
2. Entwicklung einer Chat-Anwendung <ul style="list-style-type: none"> • Nutzung der Datenstruktur List 		
3. Entwicklung einer beliebigen Client-Server-Anwendung		
4. Übung und Vertiefung der OOM / OOP		

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik 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 Schülerinnen und Schüler
3. Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
4. Medien und Arbeitsmittel sind schülernah gewählt.
5. Die Schülerinnen und Schüler erreichen einen Lernzuwachs.
6. Der Unterricht fördert eine aktive Teilnahme der Schülerinnen und Schüler
7. Der Unterricht fördert die Zusammenarbeit zwischen den Schülerinnen und Schülern und bietet ihnen Möglichkeiten zu eigenen Lösungen.
8. Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schülerinnen und Schüler
9. Die Schülerinnen und Schüler 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 erkennen.
18. Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
19. Der Unterricht ist handlungsorientiert, d. h. projekt- und produktorientiert angelegt.
20. Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
21. Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbewertung

2.3.1 Transparenz der Leistungsbeurteilung

Schulische Leistungsbewertung steht im Spannungsfeld pädagogischer und gesellschaftlicher Zielsetzung.

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

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

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

Die gesellschaftliche Funktion von Noten zu erfüllen ist der Schule aufgegeben. Noten entscheiden mit über Schullaufbahnen, Versetzungen und Abschlüsse. Zeugnisse sind mit entscheidender Parameter bei der Zuteilung von Berufs- und Lebenschancen. Daraus erwachsen für die Beurteilenden eine besondere Verantwortung und die Pflicht einer größtmöglichen Objektivität bei der Notenfindung.

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

2.3.2 Grundsätze der Leistungsbeurteilung

Es gelten folgende Grundsätze der Leistungsbewertung:

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

2.3.3 Formen der Leistungsüberprüfung

2.3.3.1 Kursarbeiten bzw. Klausuren

Kursarbeiten bzw. Klausuren dienen der schriftlichen Überprüfung der Lernergebnisse einer vorausgegangenen Unterrichtsreihe. Sie sind so anzulegen, dass Sachkenntnisse und methodische Fertigkeiten nachgewiesen werden können. Sie bedürfen einer angemessenen Vorbereitung und verlangen klare Aufgabenstellungen. Im Umfang und Anforderungsniveau sind Kursarbeiten bzw. Klausuren abhängig von den kontinuierlich ansteigenden Anforderungen entsprechend dem Lehrplan.

Es ist darauf zu achten, dass nicht nur die Richtigkeit der Ergebnisse und die inhaltliche Qualität, sondern auch die angemessene Form der Darstellung unabdingbare Kriterien der Bewertung der geforderten Leistung sind.

Am Gymnasium Lohmar werden die Kursarbeiten bzw. Klausuren in der Regel nach einem vorab festgelegten Punkteschema bewertet. Dabei ist eine glatt ausreichende Leistung bei 45% der Punktzahl erreicht worden. Die übrigen Notenstufen ergeben sich dann dadurch, dass für jede Notenstufe Intervalle der erreichten Punkte gebildet werden, die in der Regel gleich groß sind:

Note	Punkte	Prozentbereich
sehr gut plus	15	100 – 95
sehr gut	14	94 – 90
sehr gut minus	13	89 – 85
gut plus	12	84 – 80
gut	11	79 – 75
gut minus	10	74 – 70
befriedigend plus	9	69 – 65
befriedigend	8	64 – 60
befriedigend minus	7	59 – 55
ausreichend plus	6	54 – 50
ausreichend	5	49 – 45
ausreichend minus	4	44 – 39
mangelhaft plus	3	38 – 33
mangelhaft	2	32 – 27
mangelhaft minus	1	26 – 20
ungenügend	0	19 – 0

In der Sekundarstufe II wird spätestens in der Abiturvorklausur die im Zentralabitur gemäß der oben aufgeführten Tabelle vorgegebene Zuordnung der erreichten Punkte (maximale Punktzahl: 100 im GK) zur Note als Grundlage der Notenfindung genutzt. Dabei werden nur volle Punkte vergeben.

Die Fachkonferenz legt die Dauer der Kursarbeiten und Klausuren fest. Am Gymnasium Lohmar gelten für die Sekundarstufe II folgende Regelungen:

	1. Klausur, 1. HJ	2. Klausur 1. HJ	1. Klausur 2. HJ	2. Klausur, 2. HJ
EF	90 min	---	90 min	---
Q1 GK	90 min	90 min	135 min	135 min
Q2 GK	135 min	135 min	180 min	---

In der Qualifikationsphase I kann die erste Klausur im 2. Halbjahr durch eine Facharbeit ersetzt werden.

2.3.3.2 Mitarbeit im Unterricht

Der Beurteilungsbereich „Mitarbeit im Unterricht“ erfasst die Qualität und Kontinuität der Beiträge, die die Schülerinnen und Schüler im Unterricht erbringen. Diese Beiträge sollen unterschiedliche mündliche und schriftliche Formen in enger Bindung an die Aufgabenstellung, die inhaltliche Reichweite und das Anspruchsniveau der jeweiligen Unterrichtseinheit umfassen.

Bei den mündlichen Leistungen im Unterricht sind zu bewerten:

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Mitarbeit in Partner- und Gruppenarbeitsphase
- Führen eines Lerntagebuchs

Neben der Richtigkeit, Vollständigkeit und Komplexität der Gedankengänge sind die der Altersstufe angemessene sprachliche Darstellung und die Verwendung der Fachsprache von Bedeutung.

Bei der Unterrichtsgestaltung sind den Schülerinnen und Schülern hinreichend Möglichkeiten zur Mitarbeit zu eröffnen, z.B. durch

- praktische Leistungen am Computer als Werkzeug im Unterricht,
- Protokolle und Referate,
- Führen eines Lerntagebuchs,
- Projektarbeit (oft in Form von Gruppenarbeit),
- Lernerfolgsüberprüfungen und schriftliche Übungen.

2.3.3.3 Individuelle Förderung

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

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

Leistungsstärkere Schülerinnen und Schüler können ihr Wissen anhand von vertiefenden Problemstellungen erweitern.

2.3.3.4 Bildung der Zeugnisnote

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

2.4 Lehr- und Lernmittel

Eingesetzte Lehrbücher und Arbeitsmaterialien:

- Skripte und Arbeitsblätter (insbesondere Leitprogramm in EF)
- Arbeitsblätter und Programmvorlagen und auf der schuleigenen Lernplattform Moodle
- Lehrbuch: Band 1 -Informatik - Lehrwerk für die gymnasiale Oberstufe - Schöningh Verlag

Eingesetzte Software (jeweils in der aktuellen Version):

- Java SDK
- BlueJ
- Greenfoot
- Java-Editor
- Netbeans
- JFlap
- xampp
- Cryptool
- Weitere Demonstrationsprogramme

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Nutzung außerschulischer Lernorte

- Besuch der SchülerKrypto der Universität Bonn in der EF

4 4 Qualitätssicherung und Evaluation

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

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

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Jeweils vor Beginn eines neuen Schuljahres, d.h. erstmalig nach Ende der Einführungsphase im Sommer 2015 werden in einer Sitzung der Fachkonferenz für die nachfolgenden Jahrgänge zwingend erforderlich erscheinende Veränderungen diskutiert und ggf. beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können.

Nach Abschluss des Abiturs 2017 wird eine Arbeitsgruppe aus den zu diesem Zeitpunkt in der gymnasialen Oberstufe unterrichtenden Lehrkräften auf der Grundlage ihrer Unterrichtserfahrungen eine Gesamtsicht des schulinternen Curriculums vornehmen und eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.