



Informatik

am Conrad von Soest Gymnasium

2000...



Westfälischer Friedenspreis

2007...



Bürgerpreis der Stadt Soest

2009...



Partnerschule des Leistungssports

2009...



Europäischer Jugend-Karlspreis

2016...



MINT-Freundliche Schule

Inhaltsverzeichnis

Inhalt

1	Curriculum Sekundarstufe I.....	3
1.1	Fachdidaktische/ -methodische Grundsätze.....	3
1.2	Unterrichtsvorhaben/ Kompetenzerwartungen	8
1.2.1	Klasse 7.....	8
1.2.2	Klasse 8.....	14
1.2.3	Klasse 9.....	20
1.3	Qualitätssicherung und Evaluation	24
2	Curriculum Sekundarstufe II.....	24
2.1	fachdidaktische/-methodische Grundsätze.....	24
2.2	Unterrichtsvorhaben/Kompetenzerwartungen	26
2.2.1	Einführungsphase.....	26
2.2.2	Qualifikationsphase.....	36
2.3	Qualitätssicherung und Evaluation	61
3	Leistungskonzept	62
3.1	Grundsätze der Leistungsbewertung und Leistungsrückmeldung	62
3.2	Überprüfung der sonstigen Mitarbeit.....	63
3.3	Überprüfung der schriftlichen Leistung.....	64
3.4	Qualitätssicherung und Evaluation	66
4	Methodenkonzept	67
5	Medienkonzept.....	68



1 Curriculum Sekundarstufe I

1.1 Fachdidaktische/ -methodische Grundsätze

Die Verteilung der Wochenstundenzahlen in der Sekundarstufe I und II ist wie folgt:

Jg.	Fachunterricht 7
7	IT (2)
	Fachunterricht von 8 bis 9
8	Informatik Differenzierung (2)
9	Informatik Differenzierung (2)
	Fachunterricht in der EF und in der Q-Phase
10	Informatik (3)
11	Informatik (3/5)
12	Informatik (3/5)

Die Unterrichtstaktung an der Schule folgt einem 90 Minutenraster.

Informatische Inhalte und Fragestellungen werden am Conrad-von-Soest-Gymnasium in der Jahrgangsstufe 7 verbindlich im IT-Fach, im Wahlpflichtbereich II der Jahrgangsstufen 8 und 9 im Fach Informatik (mit 2 Schulstunden pro Halbjahr) und in der Gymnasialen Oberstufe in Grund- und Leistungskursen behandelt.

In nahezu allen Unterrichtsvorhaben wird den Schülerinnen und Schülern die Möglichkeit gegeben, am PC zu arbeiten; Insgesamt werden überwiegend kooperative, die Selbstständigkeit des Lerners fördernde Unterrichtsformen genutzt, sodass ein individualisiertes Lernen in der Sekundarstufe II kontinuierlich unterstützt wird. Hierzu eignen sich besonders Doppelstunden. Um die Qualität des Unterrichts nachhaltig zu entwickeln, vereinbart die Fachkonferenz vor Beginn jedes Schuljahres neue unterrichtsbezogene Entwicklungsziele. Aus diesem Grunde wird am Ende des Schuljahres überprüft, ob die



bisherigen Entwicklungsziele weiterhin gelten und ob Unterrichtsmethoden, Diagnoseinstrumente und Fördermaterialien ersetzt oder ergänzt werden sollen.

Der Unterricht des Wahlschwerpunktes Informatik wird auf Grundlage des gültigen Kernlehrplans erteilt. Schwerpunkte sind u.a. der sachgerechte Umgang mit Office-Programmen, Grundlagen der Algorithmik mithilfe von didaktischen Lernumgebungen. Zusätzlich werden die Robotermodelle in der AG Robotics behandelt.

Die Unterrichtsinhalte werden vor allem im Hinblick auf die Erziehungs- und Bildungsgrundsätze der Schule ausgewählt, die im Schulprogramm festgeschrieben sind. Hierzu gehören neben der Beherrschung fachlicher Standardqualifikationen vor allem auch grundlegende Kompetenzen und Schlüsselqualifikationen für das Berufsleben. Zu nennen sind hier Kommunikationsfähigkeit, Teamfähigkeit, selbstständiges Lernen und Lern- und Leistungsbereitschaft. Grundlagen hierzu werden im Schwerpunkt „Lernen lernen - Methodentraining“ gelegt und im Wahlpflichtfach Informatik aufgegriffen und vertieft.

Die inhaltliche Gestaltung der Unterrichtsvorhaben erfolgt zeitweilig in Form von Projekten. Dabei spielen Informationstechnologien eine zentrale Rolle.

Hinsichtlich eines fächerverbindenden Lernens stellt das Fach Informatik insbesondere zu den Fachkonferenzen Mathematik und Deutsch einen engen Bezug her. In den Unterrichtsvorhaben zum Themenbereich Tabellenkalkulation werden mathematische Inhalte aus den Bereichen Prozent- und Zinsrechnung, Statistik und Diagramminterpretation aufgegriffen und vertieft. Die in der Jahrgangsstufe 9 im Fach Deutsch erarbeiteten Bewerbungen und Lebensläufe werden durch die Unterrichtsvorhaben „Textverarbeitung“ ebenso unterstützt wie die Verschriftlichung von Referaten. Die Vermittlung des sinnvollen Umgangs mit Recherchertools im Internet unterstützt die Erarbeitung von Referaten in anderen Fächern genauso wie das Erlernen eines sicheren Umgangs mit einer Präsentationssoftware.

Die Fachgruppe Informatik des Conrad-von-Soest Gymnasiums bildet an dieser Schule eine eigene Fachkonferenz. Alle verwalteten Daten - wie Protokolle, Absprachen und Unterrichtsmaterialien - werden zentral gespeichert und sind somit allen Kolleginnen und Kollegen jederzeit zugänglich.

Die Entwicklung des schulinternen Lehrplans ist Ergebnis der Fachschaftsarbeit. Die jahrgangsübergreifende Darstellung der Themenbereiche erfolgte arbeitsteilig und wurde in der Fachkonferenz diskutiert, zusammengeführt und abschließend als verbindlich beschlossen.

Die konkreten Unterrichtsvorhaben mit den benötigten Unterrichtsmaterialien werden von den Fachlehrerinnen und -lehrern im Team unter Berücksichtigung der speziellen Vorgaben durch die Lerngruppen erarbeitet. Die Unterrichtsvorhaben und deren inhaltliche Schwerpunkte sind verbindlich festgelegt. Das gesamte Lehrmaterial wird den Fachkollegen an zentraler Stelle zur Verfügung gestellt. Nach Möglichkeit werden die Klassenarbeiten parallel geschrieben und nach gemeinsamen Vorgaben bewertet.



Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des Lehrplans durch die Fachkonferenz Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und Entwicklung des Unterrichts dar.

Der Informatikunterricht wird zurzeit von vier Lehrkräften unterrichtet, denen zwei Computerräume zur Verfügung stehen. Die Computerräume sind mit jeweils 15 Computerarbeitsplätzen für die Schülerinnen und Schüler, einem Computerarbeitsplatz für die Lehrkraft, einem Laserdrucker zur Ausgabe von Schülerarbeiten sowie einem fest installierten Beamer ausgestattet. Alle Computerarbeitsplätze sind an das schulinterne Rechnernetz des Conrad-von-Soest Gymnasiums angeschlossen. Die Lehrkräfte sowie die Schülerinnen und Schüler verfügen über individuelle Zugangsdaten zum zentralen Server der Schule und können somit alle Computerarbeitsplätze für den Zugriff auf ihre eigenen Daten, zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden.

Mit dem Schulträger findet regelmäßig ein Austausch hinsichtlich der Weiterentwicklung der schulischen IT-Infrastruktur statt.

Um allen Lernenden optimale Fortschritte zu ermöglichen, werden die Heterogenität der Lerngruppe und der unterschiedliche Kenntnisstand der Schülerinnen und Schüler berücksichtigt. Zur individuellen Förderung im Rahmen von ‚Innerer Differenzierung‘ und ‚Individualisierung‘ wurden Materialien erarbeitet, um im Unterricht leistungsstärkere Schülerinnen und Schüler gezielt zu fördern. Unter anderem wurden hierzu zusätzliche Aufgaben auf einem höheren Niveau konzipiert. Darüber hinaus unterstützen sich die Schülerinnen und Schüler insbesondere bei der Arbeit am Computer gegenseitig. Vor den jeweiligen Klassenarbeiten erhalten die Schülerinnen und Schüler eine Übersicht zu den Kompetenzen und Kenntnissen, die sie für ein erfolgreiches Bestehen der Klassenarbeit benötigen. Sie können damit ihre Stärken und Schwächen einschätzen und erhalten dazu ggf. spezielle Übungsphasen bzw. Übungsmaterial.

Es finden regelmäßig eintägige Exkursionen in das Heinz-Nixdorf-Forum nach Paderborn statt.



Fachliche Grundsätze:

Der Unterricht orientiert sich am aktuellen Stand der Informatik.

Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.

Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch Informatiksysteme aus der Berufs- und Lebenswelt eingesetzt.

Der Unterricht ist problemorientiert und knüpft an die Interessen und Erfahrungen der Schülerinnen und Schüler an.

Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert. Dazu beschäftigen sich die Schülerinnen und Schüler auch mit aktuellen Informatiksystemen und deren weiterer Entwicklung, soweit diese absehbar ist.

Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.

Der Unterricht fördert vernetzendes Denken und wird deshalb phasenweise fach- und lernbereichsübergreifend ggf. auch projektartig angelegt.

Der Unterricht beinhaltet reale Begegnung sowohl an inner- als auch an außerschulischen Lernorten wie z.B. den Kooperationsbetrieben.

Der Unterricht leistet einen wichtigen Beitrag zur Vorbereitung auf Ausbildung und Beruf und zeigt informatikaffine Berufsfelder auf.



Entscheidungen zu fach- und unterrichtsübergreifenden Fragen:

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für einige zentrale Schwerpunkte entschieden, die vorrangig zu folgenden fach- und unterrichtsübergreifenden Entscheidungen geführt haben.

Zusammenarbeit mit anderen Fächern

Kooperation mit den Fächern Deutsch und Mathematik findet bereits statt. Für das Fach Mathematik werden die Inhalte Prozentrechnung und Diagramme mit Hilfe der Tabellenkalkulation vertieft.

Ziel ist es, in Zukunft weitere Synergieeffekte mit anderen Fächern zu erreichen.

Zur Kooperation mit den **Naturwissenschaften** bieten sich die Themenbereiche Farbenlehre und Modellbildung und Simulation (z.B. radioaktiver Zerfall) an.

Mit den **Gesellschaftswissenschaften** gibt es diverse Anknüpfungspunkte bei der Auswertung von Umfragen oder gezielter Anwendung von Präsentationstechniken. Schülerinnen und Schüler aus dem WP-Fach Informatik können einerseits erlerntes Wissen an Praxisbeispielen anwenden und umgekehrt wird der Informatik-Unterricht durch diese Praxisbeispiele bereichert. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden. Da im Inhaltsfeld „Informatik, Mensch und Gesellschaft“ auch gesellschaftliche und ethische Fragen im Unterricht angesprochen werden, soll eine mögliche Zusammenarbeit mit dem Fach Gesellschaftslehre in einer gemeinsamen Fachkonferenz ausgelotet werden.

Projekttag

Bei Projekttagen, die am Conrad von Soest Gymnasium stattfinden, bietet die Fachkonferenz Informatik in diesem Zusammenhang mindestens zwei Projekte für Schülerinnen und Schüler der Jahrgangsstufen 5 bis 9 an.

Unterrichtsgänge

Um den Praxisbezug des Faches zu verdeutlichen, wird ein jährlicher Unterrichtsgang angestrebt, der einen direkten Bezug zu einem aktuellen Unterrichtsvorhaben hat. Mögliche Ziele sind die DASA, die FH Dortmund und das Heinz Nixdorf MuseumsForum. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

Fortbildungskonzept

Kollegiumsintern führen Kolleginnen und Kollegen zudem regelmäßig im Rahmen des schulischen Gesamt-Fortbildungskonzepts einmal im Jahr Fortbildungen zu speziellen Themen durch, z.B. zum Lernen mit digitalen Medien, dem Umgang mit neuen Lehrplänen, zum Umgang mit neuen technischen Geräten etc. Auf diese Weise bilden sich die Kolleginnen und Kollegen im Fachbereich Informatik fort und konzipieren weitere konkretisierte Unterrichtsvorhaben und Materialien, die dann im Fachunterricht Informatik oder auch fachübergreifend eingesetzt werden können.



1.2 Unterrichtsvorhaben/ Kompetenzerwartungen

1.2.1 Klasse 7

Jahrgangsstufe 7

Unterrichtsvorhaben I:

Thema: Wie funktioniert unser Schulnetz?

Kompetenzen:

Darstellen und Interpretieren

- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen

Zeitbedarf: 8 Std.



Unterrichtsvorhaben II:

Thema: Schreib mal wieder! - Wie werden Texte mit dem Computer ansprechend gestaltet?

Kompetenzen:

- Argumentieren
- Modellieren und Implementieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Information und Daten
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Erfassung, Verarbeitung und Verwaltung von Daten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen

Zeitbedarf: 16-18 Std.



Unterrichtsvorhaben III:

Thema: *Jetzt wird es bunt – Graphiken und Bilder mit dem Computer*

Kompetenzen:

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

Inhaltsfelder:

- Information und Daten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen

Zeitbedarf: 14 Std.



Unterrichtsvorhaben IV:

Thema: *Vom Programmbaustein zum Computerspiel – wie programmiert man einfache Animationen und Spiele?*

Kompetenzen:

- Argumentieren
- Modellieren und Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Algorithmen
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Algorithmen und algorithmische Grundkonzepte
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen

Zeitbedarf: 20 Std.



Unterrichtsvorhaben V:

Thema: Das weltweite Datennetz – ein Geheimnis?

Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 14 Std.



Unterrichtsvorhaben VI:

Thema: *Wo spielen Computer in Alltagsgeräten eine Rolle?*

Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Sprachen und Automaten
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Erfassung, Verarbeitung und Verwaltung von Daten
- Formale Sprachen und einfache Automaten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen

Zeitbedarf: 6-8 Std.

Summe 80 Stunden



1.2.2 Klasse 8

Jahrgangsstufe 8

Unterrichtsvorhaben I:

Thema: Wir analysieren Webseiten und erstellen eigene Präsentationen für das Internet

Kompetenzen:

1. Argumentieren
2. Modellieren und Implementieren
3. Darstellen und Interpretieren

Inhaltsfelder:

4. Information und Daten
5. Sprachen und Automaten
6. Informatiksysteme
7. Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Formale Sprachen und einfache Automaten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 18 Std.



Unterrichtsvorhaben II:

Thema: *Ab in die Zelle – Berechnungen und Darstellung von Daten mit der Tabellenkalkulation*

Kompetenzen:

- Argumentieren
- Modellieren und Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 14 Std.



Unterrichtsvorhaben III:

Thema: *Etwas fürs Auge – wie nutze ich Präsentationsprogramme zur Unterstützung meiner Vorträge?*

Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informationen und Daten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Erfassung, Verarbeitung und Verwaltung von Daten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 14 Std.



Unterrichtsvorhaben IV:

Thema: *Mein digitaler Fußabdruck – wo hinterlasse ich Daten und was kann daraus geschlossen werden?*

Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 10 Std.



Unterrichtsvorhaben V:

Thema: *Computer in der Arbeitswelt – Fluch oder Segen?*

Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Kommunizieren und Kooperieren

Inhaltsfelder:

- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Anwendung von Informatiksystemen
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 12 Std.



Unterrichtsvorhaben VI:

Thema: Daten auf Wanderschaft – wie kommunizieren Computer?

Kompetenzen:

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

Inhaltsfelder:

- Information und Daten
- Algorithmen
- Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Algorithmen und algorithmische Grundkonzepte
- Formale Sprachen und einfache Automaten
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 12 Std.

Summe 80 Stunden



1.2.3 Klasse 9

Jahrgangsstufe 8

Unterrichtsvorhaben I:

Thema: Geheim ist geheim? Sichere Kommunikation mit Kryptographie

Kompetenzen:

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

Inhaltsfelder:

- Information und Daten
- Algorithmen
- Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten
- Formale Sprachen und einfache Automaten
- Anwendung verschiedener Informatiksysteme
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 16 Std.



Unterrichtsvorhaben II:

Thema: Der Blick in die Glaskugel - Simulation und Prognose mit Hilfe einer Tabellenkalkulation

Kompetenzen:

- Argumentieren
- Modellieren und Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Algorithmen
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten
- Anwendung verschiedener Informatiksysteme
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 14 Std.



Unterrichtsvorhaben III:

Thema: Jäger und Sammler – Wie werden Datensammlungen systematisch angelegt und verwaltet?

Kompetenzen:

- Argumentieren
- Modellieren und Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Algorithmen
- Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten
- Algorithmen mit den Algorithmischen Grundkonzepten entwerfen, darstellen und realisieren
- Formale Sprachen und einfache Automaten
- Aufbau und Funktionsweise von Informatiksystemen und ihren Komponenten
- Anwendung verschiedener Informatiksysteme
- Informatiksysteme im Kontext gesellschaftlicher und rechtlicher Normen
- Chancen und Risiken bei der Nutzung von Informatiksystemen

Zeitbedarf: 24 Std.



Unterrichtsvorhaben IV:

Thema: Vom Programmbaustein zum Computerspiel – wie programmiert man einfache Animationen und Spiele?

Kompetenzen:

- Argumentieren
- Modellieren und Implementieren
- Darstellen und Interpretieren

Inhaltsfelder:

- Information und Daten
- Algorithmen
- Informatiksysteme

Inhaltliche Schwerpunkte:

- Daten und ihre Codierung
- Erfassung, Verarbeitung und Verwaltung von Daten
- Algorithmen und algorithmische Grundkonzepte
- Aufbau und Funktionsweise einfacher Informatiksysteme
- Anwendung von Informatiksystemen

Zeitbedarf: 22 Std.

Summe 76 Stunden

Die Gesamtstundenzahlen im Jahrgang 9 verringern sich durch das zweiwöchige Betriebspraktikum. Dies wurde bei der Planung der Unterrichtsvorhaben berücksichtigt.



1.3 Qualitätssicherung und Evaluation

Die Fachgruppe Informatik bemüht sich um eine stete Sicherung der Qualität ihrer Arbeit. Dazu dient unter anderem die jährliche Evaluation des schulinternen Curriculums mit Hilfe einer Checkliste. Weitere anzustrebende Maßnahmen der Qualitätssicherung und Evaluation sind zum Beispiel *team teaching*, Parallelarbeiten oder gemeinsames Korrigieren. Absprachen dazu werden von den in den Jahrgängen parallel arbeitenden Kolleginnen und Kollegen zu Beginn eines jeden Schuljahres getroffen.

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 sich vor allem aus den flexiblen Variablen Schülerzahl, Fachgruppengröße, Lehr- und Lernmittelentwicklung und Abiturvorgaben ergeben.

Der Prüfmodus erfolgt jährlich. In den Dienstbesprechungen der Fachgruppe zu Schuljahresbeginn werden die Erfahrungen des vorangehenden Schuljahres gesammelt und bewertet sowie eventuell notwendige Konsequenzen formuliert. Die vorliegende Checkliste wird als Instrument einer solchen Bilanzierung genutzt. Die Ergebnisse dienen dem/der Fachvorsitzenden zur Rückmeldung an die Schulleitung und u.a. an den/die Fortbildungsbeauftragte, außerdem sollen wesentliche Tagesordnungspunkte und Beschlussvorlagen der Fachkonferenz daraus abgeleitet werden. Insgesamt dient die Checkliste über die Evaluation des aktuellen schulinternen Curriculums hinaus zur systematischen Qualitätssicherung und Qualitätsentwicklung der Arbeit der Fachgruppe.

Quelle:

<http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/englisch/hinweise-und-beispiele/schulinterner-lehrplan/schulinterner-lehrplan.html>

2 Curriculum Sekundarstufe II

2.1 fachdidaktische/-methodische Grundsätze

Für den Informatikunterricht in der Sekundarstufe II wurden am Conrad-von-Soest Gymnasium zwei Schulbücher von Schönigh (Informatik 1 – Lehrwerk für die gymnasiale Oberstufe / Informatik 2 – Lehrwerk für die gymnasiale Oberstufe) eingeführt.



- Als Werkzeuge werden Greenfoot oder GLOOP empfohlen.
- Eine Unterrichtsstunde ist 90 min lang. Die im folgenden Unterrichtsrate angegebenen Zeitangaben sind Richtwerte.
- Die Kompetenzbereiche des Kernlehrplans werden wie folgt abgekürzt:
 - Argumentieren (A)
 - Modellieren (M)
 - Implementieren (I)
 - Darstellen und Interpretieren (D)
 - Kommunizieren (K)

Die Schülerinnen und Schüler arbeiten die im Unterricht behandelten Inhalte in häuslicher Arbeit nach.

Die Fachkolleginnen und -kollegen werden zudem ermutigt, die Materialangebote des Ministeriums für Schule und Weiterbildung regelmäßig zu sichten und ggf. in den eigenen Unterricht oder die Arbeit der Fachkonferenz einzubeziehen. Die folgenden Seiten sind dabei hilfreich:

Der Lehrplannavigator:

<http://www.standardsicherung.schulministerium.nrw.de/lehrplaene/lehrplannavigator-s-ii/>

Die Materialdatenbank:

<http://www.standardsicherung.schulministerium.nrw.de/materialdatenbank/>

Die Materialangebote von SINUS-NRW:

<http://www.standardsicherung.nrw.de/sinus/>



2.2 Unterrichtsvorhaben/Kompetenzerwartungen

2.2.1 Einführungsphase

Unterrichtsvorhaben EF-I: Einführung in die Informatik

Leitfragen: Womit beschäftigt sich das Fach Informatik in der Sek. II? Was sind die Fragestellungen und Meilensteine der Informatik?

Vorhabenbezogene Konkretisierung: Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei sollen die Meilensteine in der Geschichte der Informatik und grundlegende Prinzipien wie das EVA-Prinzip und die Von-Neumann-Architektur erarbeitet werden.

Mit Hilfe eines geschichtlichen Abrisses und populärer informatischer Fallbeispiele sollen die Wechselbeziehung zwischen der Informatik und der Gesellschaft thematisiert und die Auswirkungen des Einsatzes von Informatiksystemen auf die Gesellschaft deutlich werden.

Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Das gesamte Vorhaben kann durch eine eintägige Exkursion in das Heinz-Nixdorf-Forum nach Paderborn ergänzt werden.

Zeitbedarf: ca. 10 Stunden (+ optionale Exkursion)

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder Materialien
1. Geschichte und „Geschichtchen“ der Informatik und ihre Auswirkungen auf die heutige Gesellschaft	<p>Die Schülerinnen und Schüler (SuS)...</p> <ul style="list-style-type: none"> • ...erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A) • ...bewerten anhand von Fallbeispielen die 	<ul style="list-style-type: none"> • Fallbeispiele (berühmte Softwarefehler: z.B. Scud-Rakete im Irak-



	Auswirkungen des Einsatzes von Informatiksystemen (A)	Krieg, Softwarefehler bei EC-Karten, Apollo-Mondlandung, Ariane 5)
2. Aufbau informatischer Systeme <ul style="list-style-type: none"> • Identifikation typischer Arbeitsweisen und Komponenten informatischer Systeme. • Herleitung der VN-Architektur • Identifikation des EVA-Prinzips als Prinzip der Verarbeitung von Daten und Grundlage der VN-Architektur. 	Die SuS... <ul style="list-style-type: none"> • ...beschreiben und erläutern den strukturellen Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der VN-Architektur (A) • ...nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K) 	<ul style="list-style-type: none"> • Aus den vorherigen Beispielen und Diskussionen werden das EVA-Prinzip die in der VN-Architektur vorkommenden Komponenten herausgearbeitet. • Hier ist die Vergabe von Referaten oder eine arbeitsteilige Erarbeitung möglich.
3. Der Binärcode als Grundlage der Datenverarbeitung	Die SuS... <ul style="list-style-type: none"> • ...stellen ganze Zahlen und Zeichen in Binärcodes dar (D) • ...interpretieren Binärcodes als Zahlen und Zeichen (D) 	<ul style="list-style-type: none"> • z.B. „Abenteuer Informatik“, Kapitel 5/10
4. Datenschutz	Die SuS... <ul style="list-style-type: none"> • ... nutzen im die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher , zielführend und Verantwortungsbewusst (D) 	<ul style="list-style-type: none"> • z.B. Facebook • www.klicksafe.de

Unterrichtsvorhaben EF-II: Einführung in die OOP/Informatisches Problemlösen und Algorithmen als Beschreibung von Abläufen



Leitfragen: Was sind Klassen, Objekte, Methoden, Attribute? Wie interagieren Objekte unterschiedlicher und gleicher Klassen miteinander? Wie beschreibt man in der Informatik Abläufe computernah? Was zeichnet Algorithmen aus und wie kann man sie vergleichen? Wie werden Algorithmen entwickelt, dargestellt und in der Programmiersprache Java umgesetzt?

Vorhabenbezogene Konkretisierung: Dieses Unterrichtsvorhaben beinhaltet eine Einführung in die Objektorientierte Modellierung (OOM). In Kurzprojekten werden Objekte, Klassen und ihre Beziehungen in Form von Objekt-, Entwurfs- und Implementationsklassendiagrammen dargestellt. In einem weiteren Schritt sollen Objekte miteinander interagieren können. Hierbei wird der Unterschied zwischen beiden Beziehungsarten (gerichtete Assoziation, Ist-Beziehung) erarbeitet und visualisiert.

Anhand von alltäglichen Beispielen werden der Algorithmus-Begriff und seine Eigenschaften herausgearbeitet. Dabei werden Algorithmen erstellt, implementiert und beispielsweise in Programmablaufplänen (PAP) dokumentiert. Bedeutsam sind dabei die Herausarbeitung der grundlegenden algorithmischen Strukturen (Sequenz, bedingte Verzweigung und kopfgesteuerte Schleife) und die Überprüfung der Eigenschaften von Algorithmen.

Begonnen wird mit einem einfachen Szenario. Nach der sukzessiven Einführung der zu behandelnden Kontrollstrukturen Verzweigung, Schleifen werden zunehmend komplexere Problemstellungen von den SuS erarbeitet und gelöst, wobei die Modellierung als Vorstufe der Implementation eingefordert wird.

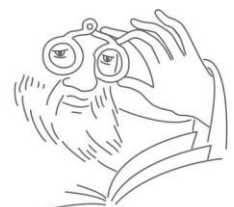
Im Anschluss werden anhand von konkreten Algorithmen die erarbeiteten Techniken geübt, Fragestellung zu Grenzen von Algorithmen thematisiert und die Idee der rekursiv formulierten Algorithmen erarbeitet. Im Anschluss arbeiten die Schüler an einem umfangreichen Projekt.

Zeitbedarf: ca. 30 Stunden + 8 Stunden Projektarbeit.

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder Materialien
<ul style="list-style-type: none"> ■ Grundlagen der OOP <ul style="list-style-type: none"> • Erarbeitung der Bedeutung von Klassen, Objekten und Methoden • Darstellung von Klassen und Objekten in UML-Notation • Benutzung von 	<p>Die SuS...</p> <ul style="list-style-type: none"> • ...ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M). • ...stellen den Zustand 	<ul style="list-style-type: none"> • Greenfoot/GLOOP • PAP-Designer, yEd, OpenOffice • Ausgehend von



<p>Objekten</p> <ul style="list-style-type: none"> ■ Algorithmik <ul style="list-style-type: none"> ■ Alltagsbeispiele zu Algorithmen ■ Erlernen grundlegender Algorithmen mit Kontrollstrukturen ■ Erarbeitung des Unterschieds zwischen Aufträgen und Anfragen anhand von Anwendungsbeispielen 	<p>eines Objekts dar (D).</p> <p>Die SuS...</p> <ul style="list-style-type: none"> • ...kennen den Begriff „Algorithmus“ und seine Eigenschaften • ...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). • ...testen Programme schrittweise anhand von Beispielen (I) • ...dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D). 	<p>den berühmten Softwarefehlern und mit Hilfe alltäglicher Beispiele wird der Algorithmus-Begriff entwickelt und an Beispielen überprüft, geschärft und eingeübt.</p> <ul style="list-style-type: none"> • Mögliche Beispiele: Papierflieger (Sequenz), Kochrezept (Sequenz und bedingte Verzweigung) ... • Material: PAP-Designer
<ul style="list-style-type: none"> ■ Objektbeziehungen modellieren und umsetzen <ul style="list-style-type: none"> ■ Analyse von Problemstellungen, bei denen die Interaktion von Objekten notwendig ist ■ Darstellung der Beziehungen in Form von Klassendiagrammen ■ Erarbeitung, Programmierung und Bewertung verschiedener Umsetzungsmöglichkeiten der Beziehungen 	<p>Die SuS...</p> <ul style="list-style-type: none"> • ...analysieren und erläutern eine objektorientierte Modellierung (A) • ...modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M). • ...modellieren Klassen unter Verwendung von 	<ul style="list-style-type: none"> • Euklidischer Algorithmus: Darstellung eines iterativen mathematischen Algorithmus mit Hilfe eines PAPs (Bedingte Verzweigung, kopfgesteuerte Schleife) und ugs. Beschreibung • Labyrinth-Problem: Die Rechte-Hand-



<ul style="list-style-type: none"> ■ Einführung des Variablenkonzepts/Datentypen <ul style="list-style-type: none"> ■ Problemorientierte Erarbeitung der Benutzung von Variablen als universellem Speicher. ■ Benutzung der Datentypen Integer, String, Char, Boolean. ■ Vertiefung der Inhalte anhand eines weiteren Anwendungsbeispiels 	<p>Vererbung (M).</p> <ul style="list-style-type: none"> • ...stellen die Kommunikation zwischen Objekten grafisch dar (M). • ...stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D). • ...implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I). <p>Die SuS...</p> <ul style="list-style-type: none"> • ...testen Programme schrittweise anhand von Beispielen (I). • ...nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K). 	<p>Regel Darstellung in Umgangssprache und im PAP / Testen von Algorithmen / Übertragung auf vergleichbare Labyrinth / Grenze des Algorithmus (Problemklasse)</p> <ul style="list-style-type: none"> • Türme von Hanoi: Herausarbeiten von Regeln und Fragestellung aus der Geschichte, Reduzierung, praktisches Ausprobieren, Darstellung als Algorithmus, Zeitproblematik (Grenze der Umsetzung des Algorithmus) Rekursive Darstellung; Eigenschaften von Rekursionen (Selbstaufruf mit Laufbedingung bei Verkleinerung des Problems)
--	--	--



Unterrichtsvorhaben EF-III: Suchen und Sortieren

Leitfragen: Wie funktioniert die effektive Suche nach Objekten? Wie funktioniert das Sortieren von Objekten? Welche Algorithmen gibt es und wie effizient sind diese?

Vorhabenbezogene Konkretisierung: Anhand von alltäglichen Beispielen wird das Suchen nach Objekten motiviert. Dabei wird zunächst die iterative Suche betrachtet. Es soll deutlich werden, dass ein Objekt wesentlich schneller gefunden werden kann, wenn eine Vorsortierung der zu durchsuchenden Objekte vorliegt. Daraus soll der Algorithmus der binären Suche entwickelt werden. Die benötigte Vorsortierung motiviert die Betrachtung von Sortieralgorithmen. Als einfache Algorithmen wird hier beispielhaft der Selection- oder Bubblesort betrachtet. Dazu gehört die Modellierung, Implementierung (in einem vorgefertigtem Szenario) und Untersuchung auf Effizienz, die sich auf die Anzahl der Vergleiche und Vertauschungen beschränkt. Die Algorithmen in diesem Unterrichtsvorhaben werden in Programmablaufplänen modelliert.

Zeitbedarf: ca. 12 Stunden

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<ul style="list-style-type: none"> ■ Suchalgorithmen (iterative und binäre Suche) ■ Einfache Sortieralgorithmen (Selection- / Bubblesort) ■ (eindimensionales) Array, for-Schleife 	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D) • entwerfen einen weiteren Algorithmus zum Sortieren (M) • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeitaufwand und Speicherplatzbedarf. (A) <p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> • analysieren und 	<ul style="list-style-type: none"> • Suchen in Alltagsbeispielen (z.B.: CD-Regal, Suche der größten/kleinsten Zahl) • Betrachtung der Effizienz der iterativen Suche • Optimierung der iterativen Suche zur binären Suche. (Die Vorsortierung als Bedingung) • Beschreibung der Algorithmen in PAPs • Umsetzung des Bubblesorts oder Selectionsorts • Zählen von



	<p>erläutern Algorithmen und Programme (A),</p> <ul style="list-style-type: none">• analysieren und erläutern einfache Algorithmen und Programme (A),• entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M),• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen• Stellen lineare und nicht lineare Strukturen grafisch dar und erläutern ihren Aufbau (D)	<p>Zugriffen / Vergleichen / (Dreieckstausch)</p>
--	--	---



Unterrichtsvorhaben EF-IV: Softwareentwicklung

Thema: Softwareentwicklung

Leitfragen: Wie wird Software entwickelt? Welche Phasen durchläuft Software während der Entwicklung? Welche Bereiche werden strikt voneinander getrennt (MVC-Konzept)?

Vorhabenbezogene Konkretisierung: Schülerinnen und Schüler durchlaufen in diesem UV einen vollständigen Softwareentwicklungsprozess (z.B. Spiel „Verflixte 7“).

Im Rahmen der Unterrichtsreihe erlernen sie die einzelnen Phasen Objektorientierte Analyse (OOA), Objektorientiertes Design (OOD) bis hin zur Objektorientierten Programmierung (OOP). In der OOA-Phase werden beteiligte Objekte mit ihren Fähigkeiten und Eigenschaften beispielsweise mit Hilfe der Substantiv-Verb Methode, Use-Case Diagramme oder der CRC-Methode identifiziert. Das OOD dient der Konkretisierung der OOA und Implementationsdiagramme werden erstellt.

Der letzte Schritt ist die Umsetzung des Projekts/Modells mit Java einschließlich der Gestaltung einer grafischen Oberfläche. Das System wird getestet und bei Bedarf verbessert, optional unter der Verwendung weiterer Diagramme. Im Anschluss arbeiten die Schüler an einem umfangreichen Projekt.

Zeitbedarf: 20 Stunden + 10 Stunden Projektarbeit

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien oder Materialien
<p>a) OOA</p> <p>Die OOA dient der Untersuchung der Aufgabenstellung und führt zu einem ersten UML-Modell. Die SuS benutzen dazu die Fachmethoden: Use-Case Diagramme, Substantiv Verb Methode, CRC Karten</p> <p>b) OOD</p> <p>Die OOD dient der Konkretisierung und führt zu einem Implementationsdiagramm</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 	<p>Use-Case-Diagramme, Substantiv-Verb-Methode, CRC-Karten, Sequenzdiagramme, BlueJ (GUI Editor)</p>



<p>c) OOP</p> <p>Mit BlueJ wird das Modell implementiert und unter anderem mit Hilfe des Debuggers und von Sequenzdiagrammen auf seine Lauffähigkeit getestet.</p> <p>d) MVC Konzept</p> <p>Das Modell wird mit Hilfe des Java-Editors um eine einfache grafische Oberfläche erweitert.</p>	<p>Klassen unter Verwendung von Vererbung (M),</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),• stellen die Kommunikation zwischen Objekten grafisch dar (M),• stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D),• implementieren Klassen in einer Programmiersprache (I).• analysieren und erläutern einfache Algorithmen und Programme (A),• entwerfen einfache Algorithmen und stellen sie	
---	---	--



	<p>umgangssprachlich und grafisch dar (M),</p> <ul style="list-style-type: none">• implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I),• nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D).• implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I),• interpretieren Fehlermeldungen und korrigieren den Quellcode (I).• dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).• analysieren und erläutern eine objektorientierte Modellierung (A)	
--	--	--

Summe Einführungsphase: 90 Stunden



2.2.2 Qualifikationsphase

Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase - Q1

Qualifikationsphase – Q1	
<p><u>Unterrichtsvorhaben Q1-I</u></p> <p>Thema: Wiederholung und Vertiefung der objektorientierten Modellierung</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">- Modellieren- Darstellen und Interpretieren- Implementieren- Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">- Daten und ihre Strukturierung- Algorithmen- Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">- Objekte und Klassen- Wirkung der Automatisierung <p>Zeitbedarf: ca. 14 Stunden</p>	<p><u>Unterrichtsvorhaben Q1-II</u></p> <p>Thema: Organisation und Verarbeitung von und Implementierung von Anwendungen mit Datenstrukturen</p> <p>Zentrale Kompetenzen:</p> <ul style="list-style-type: none">- Modellieren- Implementieren- Darstellen und Interpretieren- Argumentieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none">- Daten und ihre Strukturierung- Algorithmen <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none">- Objekte und Klassen- Syntax und Semantik einer Programmiersprache- Analyse, Entwurf und Implementierung von Algorithmen- Algorithmen in ausgewählten informatischen Kontexten- Sortieralgorithmen <p>Zeitbedarf: ca. 20 Stunden</p>



Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

Zentrale Kompetenzen:

- Argumentieren
- Darstellen und Interpretieren
- Modellieren
- Implementieren

Inhaltsfelder:

- Algorithmen
- Formale Sprachen und Automaten

Inhaltliche Schwerpunkte:

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

Zeitbedarf: ca. 20 Stunden

Unterrichtsvorhaben Q1-IV

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

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: ca. 20 Stunden

Summe Qualifikationsphase 1: ca. 74 Stunden



Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase – Q2

Qualifikationsphase – Q2	
<p><u>Unterrichtsvorhaben Q2-I</u> Thema: Automaten und formale Sprachen Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Modellieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Formale Sprachen und Automaten - Informatiksysteme <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Syntax und Semantik einer Programmiersprache - Endliche Automaten - Grammatiken regulärer Sprachen - Möglichkeiten und Grenzen von Automaten und formalen Sprachen <p>Zeitbedarf: ca. 20 Stunden</p>	<p><u>Unterrichtsvorhaben Q2-II</u> Thema: Aufbau von und Kommunikation in Netzwerken Zentrale Kompetenzen:</p> <ul style="list-style-type: none"> - Argumentieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren <p>Inhaltsfelder:</p> <ul style="list-style-type: none"> - Informatiksysteme - Informatik, Mensch und Gesellschaft <p>Inhaltliche Schwerpunkte:</p> <ul style="list-style-type: none"> - Einzelrechner und Rechnernetzwerke - Sicherheit - Nutzung von Informatiksystemen, Wirkungen der Automatisierung <p>Zeitbedarf: ca. 16 Stunden</p>



Unterrichtsvorhaben Q2-III

Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

Zentrale Kompetenzen:

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

Inhaltsfelder:

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

Inhaltliche Schwerpunkte:

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

Zeitbedarf: ca. 20 Stunden

Summe Qualifikationsphase 2: ca. 56 Stunden



Unterrichtsvorhaben Q1-I

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

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

Vorhabenbezogene Konkretisierung:

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der 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.

Zeitbedarf: ca. 14 Stunden



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder 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), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D), - untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A), - untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Inte- 	<p>2.1 Die Welt ist voller Objekte</p> <p>Projekteinstieg: Klassentwurf – step by step</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>	<p>- stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D),</p> <p>- dokumentieren Klassen (D),</p> <p>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</p> <p>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</p> <p>- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Inte-</p>	<p>2.2 Gut geplant – Klassentwurf</p> <p>2.3 Vererbungshierarchien nutzen</p>
<p>3. Mensch und Technik</p> <p>a) Verantwortung von Informatikern</p> <p>b) Automatisierung des Alltags durch Informatik</p>	<p>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</p> <p>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</p> <p>- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Inte-</p>	<p>Die digitale Welt 001 - Mensch und Technik</p>
<p>4. Übung und Vertiefung der OOM / OOP</p>	<p>- stellen die Kommunikation zwischen Objekten grafisch dar (D),</p> <p>- untersuchen und bewerten anhand von Fallbeispielen Auswirkungen des Einsatzes von Informatiksystemen sowie Aspekte der Sicherheit von Informatiksystemen, des Datenschutzes und des Urheberrechts (A),</p> <p>- untersuchen und bewerten Problemlagen, die sich aus dem Einsatz von Informatiksystemen ergeben, hinsichtlich rechtlicher Vorgaben, ethischer Aspekte und gesellschaftlicher Werte unter Berücksichtigung unterschiedlicher Inte-</p>	<p>Aufgaben zur Prüfungsvorbereitung</p>



	ressenlagen (A).	
--	------------------	--



Unterrichtsvorhaben Q1-2

Thema:

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

Leitfragen:

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

Vorhabensbezogene Konkretisierung:

Ausgehend von Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur *Queue* erarbeitet. Die Klasse *Queue* selbst wird gemeinsam modelliert und implementiert. Darauf folgt die Erarbeitung der Struktur *Stack*, die mithilfe eines einfachen Anwendungsszenarios eingeführt wird. Hier wird die Klasse *Stack* von den Schüler/innen weitestgehend selbstständig modelliert und implementiert.

Weitere Aufgaben dienen der Vertiefung und Sicherung sowie der Erarbeitung des FIFO- und LIFO-Prinzips.

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 erarbeitet und in einem weiteren Beispiel vertieft wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: ca. 20 Stunden



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder 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) 	<p>Kapitel:</p> <ul style="list-style-type: none"> - Anforderungen an eine Datenstruktur - Speichern mit Struktur <p><i>neue Wdh. Aufgabe entwickeln, z. B. eine Chart-Top-10, eine Aufgabe mit zweidimensionalem Array (vgl. Anforderungen KLP)</i></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"> - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M), - implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) - testen und korrigieren Computerprogramme (I) - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) 	<p>Kapitel:</p> <ul style="list-style-type: none"> - Wer zuerst kommt - Objekte miteinander verketteten - Verwaltung und Inhalt - Funktionen der Queue <p>Aufgaben:</p> <ul style="list-style-type: none"> - Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist. - Erweiterte Queue Verkehrskontrolle (Vertiefung) Die Polizei kontrolliert die Fahrzeuge im Hinblick auf ihre Verkehrstauglichkeit. Für die Kontrolle werden die Fahrzeuge aus dem Verkehr gewunken. Es werden so lange Fahrzeuge kontrolliert, bis eine gewissen Menge an Verstößen vorliegt oder Autos kontrolliert wurden.
<p>3. Die Datenstruktur Stapel</p> <p>a) Erläuterung von Problem-</p>	<ul style="list-style-type: none"> - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihnen 	<p>Kapitel:</p> <ul style="list-style-type: none"> - Daten gut abgelegt – Stapel - Funktionen der Datenstruktur



<p>stellungen, 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>Stack</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</i> (<i>Palindrom</i>))</p>	<p>Aufbau (D)</p> <ul style="list-style-type: none"> - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M) - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) - dokumentieren Klassen (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I) 	<p>Stapel</p> <p>Aufgaben:</p> <ul style="list-style-type: none"> - Standardoperationen/Basiskompetenz (Stapel Münzen/ CDs) zur Umsetzung der gegebenen Funktionen der Klasse <i>Stack</i> - Biber und Teller Es gibt große und kleine Biber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl Queue als auch Stack verwendet werden. - Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.
<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>Kapitel:</p> <ul style="list-style-type: none"> - Flexibel für alle Fälle – (die) lineare Liste - Funktionen der Datenstruktur Liste <p>Aufgaben:</p> <ul style="list-style-type: none"> - LEDs - Textzeilen verarbeiten
<p>5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen</p>		<p>Aufgaben zur Prüfungsvorbereitung</p>





Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

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

Vorhabenbezogene Konkretisierung:

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

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

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

Zeitbedarf: ca. 20 Stunden



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder 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 „Teile und Herrsche“ (M), 	<p>4.1 Ohne Algorithmen läuft nichts</p> <p>4.2 Teile die Arbeit – rekursive Algorithmen</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). 	<p>4.3 Suchen – iterativ und rekursiv</p>
<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"> - 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), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I), 	<p>4.4 Sortieren – iterativ und rekursiv</p>



Unterrichtsvorhaben Q1-IV

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

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

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

Zeitbedarf: ca. 20 Stunden



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder Materialien
<p>1. Aufbau von Baumstrukturen und Grundbegriffe</p> <p>a) Erarbeitung der Begriffe Wurzel, Knoten, Blatt, Kante, Grad eines Knotens und eines Baumes, Pfad, Tiefe, Ebene, Teilbaum</p> <p>b) Aufbau und Darstellung von Baumstrukturen in verschiedenen Anwendungskontexten</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D), - erläutern Operationen dynamischer (linearer oder nichtlinearer) Datenstrukturen (A), - analysieren und erläutern Algorithmen und Programme (A), 	<p>6.1 Spielen mit Struktur – Baumstrukturen</p> <p>Projekteinstieg 1: Spielbäume</p>
<p>2. Binäre Bäume</p> <p>a) rekursiver Aufbau eines binären Baums</p> <p>b) Traversierungen (pre-, in-, postorder)</p> <p>c) Modellierung eines Binärbaums in einem Anwendungskontext mit Hilfe der Klasse BinaryTree (als Entwurfs- und Implementationsdiagramm)</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Baum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D). - 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), 	<p>6.2 Zwei Nachfolger sind genug! - Binäre Bäume</p> <p>Implementation des Projekts Ahnenbaum</p>
<p>3. Binäre Suchbäume</p> <p>a) Prinzip des binären Suchbaums, Ordnungsrelation</p> <p>b) Operationen auf dem binären Suchbaum (Suchen, Einfügen, Löschen, sortierte Ausgabe)</p> <p>c) Modellierung eines binären Suchbaums in einem Anwendungskontext mit Hilfe der Klasse BinarySearchTree (als Entwurfs- und Implementationsdiagramm) und dem Interface Item</p> <p>d) Implementation einer Anwendung der Datenstruktur binärer Suchbaum (ggf. in Teilen)</p>	<ul style="list-style-type: none"> - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - verwenden bei der Modellierung geeigneter Problemstellungen die Möglichkeiten der Polymorphie (M), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Konstruktionsstrategien „Modularisierung“ und 	<p>6.3 Wer Ordnung hält, spart Zeit beim Suchen – Binäre Suchbäume</p> <p>Projekteinstieg 2: Binäre Suchbäume</p> <p>Implementation des Projekts Benutzerverwaltung</p>



<p>4. Graphen (nur LK) - ungerichtet gewichtete Graphen - Graphalgorithmen</p>	<p>„Teilen und Herrschen“ (M), - LK: entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“, „Teile und Herrsche“ und „Backtracking“ (M)</p>	
<p>5. Übung und Vertiefungen der Verwendung von Binärbäumen oder binären Suchbäumen anhand weiterer Problemstellungen</p>	<p>- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - modifizieren Algorithmen und Programme (I), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I),</p>	<p>Aufgaben zur Prüfungsvorbereitung</p>



Unterrichtsvorhaben Q2-I

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?

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

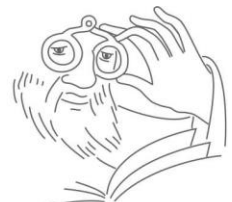
eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

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.

Zeitbedarf: ca. 20 Std.



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder 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>d) Kellerautomaten (nur LK)</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) - LK: analysieren und erläutern die Eigenschaften von Kellerautomaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A) - ermitteln die Sprache, die ein endlicher Automat akzeptiert (D) - entwickeln und modifizieren zu einer Problemstellung endliche Automaten (M) - LK: entwickeln und modifizieren zu einer Problemstellung Kellerautomaten (M) 	<p>Vom realen Automaten zum Modell Projekteinstieg: Schatzsuche Der Mealy-Automat</p> <p>Der erkennende endliche Automat</p> <p>Wort und Sprache</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"> - 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) - LK: entwickeln zur Grammatik einer kontextfreien Sprache einen zugehörigen Kellerautomaten (M) - analysieren und erläutern Grammatiken regulärer Sprachen (A) - LK: analysieren und erläutern Grammatiken kontextfreier Sprachen (A) 	<p>Grammatiken regulärer Sprachen</p>
<p>3. Grammatiken kontextfreier Sprachen (nur LK)</p>	<ul style="list-style-type: none"> - modifizieren Grammatiken regulärer Sprachen (M) - LK: modifizieren Grammatiken kontextfreier Sprachen (M) - entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M) - LK: entwickeln zu einer kontext- 	



	<p>freien Sprache eine Grammatik, die die Sprache erzeugt (M)</p> <ul style="list-style-type: none"> - entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M) - beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D) - zeigen die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang auf (A) - LK: erläutern die Grenzen endlicher Automaten und regulärer Grammatiken im Anwendungszusammenhang(A) 	
<p>4. Entwicklung von Scanner, Parser und Interpreter für eine reguläre Sprache (nur LK)</p>	<p>- LK: modellieren und implementieren Scanner, Parser und Interpreter zu einer gegebenen regulären Sprache (I)</p>	
<p>5. Übungen und Vertiefungen Verwendung endlicher Automaten und Grammatiken regulärer Sprachen</p>		<p>Aufgaben zur Prüfungsvorbereitung</p>



Unterrichtsvorhaben Q2-II

Thema: Aufbau von und Kommunikation in Netzwerken

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

Vorhabenbezogene Konkretisierung:

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

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

Zeitbedarf: ca. 16 Stunden



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder Materialien
<p>1. Technische Kommunikation als Fortführung natürlicher Kommunikation</p> <p>a) Kommunikation im Shannon-Weaver-Modell</p> <p>b) Kriterien von technischen Kommunikationsarten</p> <p>c) Die Geschichte der technischen Kommunikation</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - beschreiben und erläutern Netzwerk-Topologien, die Client-Server-Struktur und Protokolle sowie ein Schichtenmodell in Netzwerken (A), <p>- LK: analysieren und erläutern Protokolle zur Kommunikation in einem Client-Server-</p>	<p>7.1 Menschen kommunizieren – ohne und mit Technik</p> <p>Projekteinstieg: Kommunikation im Wilden Westen</p>
<p>2. Aufbau von Netzwerken und Kommunikationsregeln</p> <p>a) Das Netzwerk als Organisationsprinzip der Kommunikation und Möglichkeiten der Ausformung</p> <p>b) Geregeltete technische Kommunikation durch Protokolle in Schichtenmodellen</p>	<p>Netzwerk (A),</p> <p>- LK: entwickeln und erweitern Protokolle zur Kommunikation in einem Client-Server-Netzwerk (M).</p> <ul style="list-style-type: none"> - nutzen bereitgestellte Informatiksysteme und das Internet reflektiert zum Erschließen, zur Aufbereitung und Präsentation fachlicher Inhalte (D), 	<p>7.2 Ohne Protokoll läuft nichts - Netzwerke</p>
<p>3. Aufgabenteilung in Netzwerken durch Server und Client</p> <p>a) Aufbau und Aufgaben der Client-Server-Struktur</p> <p>b) Protokolle zwischen Client und Server</p>	<ul style="list-style-type: none"> - analysieren und erläutern Eigenschaften und Einsatzbereiche symmetrischer und asymmetrischer Verschlüsselungsverfahren (A) 	<p>7.3 Einer für alle – Client-Server-Struktur</p>
<p>4. Modellierung, Entwurf und Implementation einfacher Client-Server-Anwendungen (nur im LK)</p>		
<p>5. Kryptologie</p> <p>a) Veranschaulichen und Anwenden von symmetrischen und asymmetrischen kryptographischen Verfahren (Caesar, Vigenère, RSA)</p> <p>b) Bewertung der Verfahren hinsichtlich ihrer Sicherheit und ihrem Aufwand</p>		<p>Die digitale Welt 100 – Kryptologie</p>
<p>6. Übung und Vertiefung des Aufbaus von und der Kommunikation in Netz-</p>		<p>Aufgaben zur Prüfungsvorbereitung</p>



werken		
---------------	--	--



Unterrichtsvorhaben Q2-3

Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

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

Vorhabenbezogene Konkretisierung:

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

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

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

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

Zeitbedarf: ca. 20 Stunden



Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Mögliche Beispiele, Medien oder Materialien
<p>1. Nutzung von relationalen Datenbanken</p> <p>a) Aufbau von Datenbanksystemen und Grundbegriffe</p> <ul style="list-style-type: none"> - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen <p>b) SQL-Abfragen</p> <ul style="list-style-type: none"> - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) - Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke) <p>c) Vertiefung an einem weiteren Datenbankbeispiel</p>	<p>Die Schülerinnen und Schüler</p> <ul style="list-style-type: none"> - 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), - überführen Datenbankschemata in vorgegebene Normalformen (M), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), 	<p>Kapitel 8 Datenbanken</p> <p>8.1 Wissen speichern und verwalten – Datenbanksysteme</p> <p>8.2 Daten anordnen mit Tabellen</p> <p>Beispiel: Buchhandlung</p> <p>Redundanzen, Anomalien und Inkonsistenzen</p> <p>Projekteinstieg: Tabellen</p> <p>8.3 Daten filtern mit SQL</p> <p>8.4 Komplexe Filter</p> <p>Aufgaben</p>
<p>2. Modellierung von relationalen Datenbanken</p> <p>a) Datenbankentwurf durch ER-Diagramme</p> <ul style="list-style-type: none"> - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung 	<ul style="list-style-type: none"> - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar (D), - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D). 	<p>8.5 Datenbankentwurf</p> <p><i>Beispiel: Online-Buchhandel</i></p> <p><i>Datenanalyse und Entwurf</i></p> <p>8.6 Umsetzung des ER-Modells</p> <p><i>Entitätsmengen</i></p>



<p>einer Datenbankmodellierung</p> <p>b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf</p> <ul style="list-style-type: none"> - Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln <p>c) Normalformen</p> <ul style="list-style-type: none"> - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten) 		<p><i>m:n-Beziehungen</i></p> <p><i>1:n-Beziehungen</i></p> <p><i>1:1-Beziehungen</i></p> <p><i>Wiederaufgriff des Projekteinstiegs</i></p> <p><i>8.7 Datenbanken verbessern durch Normalformen</i></p>
<p>3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken</p>		<p>Aufgaben zur Prüfungsvorbereitung</p>

Quelle: Beispiel für ein an das Lehrwerk Informatik 2 angepasstes Unterrichtsvorhaben für einen Grundkurs als zentraler Bestandteil eines schulinternen Lehrplans zum Kernlehrplan für die gymnasiale Oberstufe, Schöningh Verlag, http://files.schulbuchzentrum-online.de/onlineanhaenge/files/978-3-14-037127-8_konkretisierte_unterrichtsvorhaben_gk.doc



2.3 Qualitätssicherung und Evaluation

Die Fachgruppe Informatik bemüht sich um eine stete Sicherung der Qualität ihrer Arbeit. Dazu dient unter anderem die jährliche Evaluation des schulinternen Curriculums mit Hilfe einer Checkliste. Weitere anzustrebende Maßnahmen der Qualitätssicherung und Evaluation sind zum Beispiel gegenseitiges Hospitieren, *team teaching*, Parallelarbeiten oder gemeinsames Korrigieren. Absprachen dazu werden von den in den Jahrgängen parallel arbeitenden Kolleginnen und Kollegen zu Beginn eines jeden Schuljahres getroffen.

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 sich vor allem aus den flexiblen Variablen Schülerzahl, Fachgruppengröße, Lehr- und Lernmittelentwicklung und Abiturvorgaben ergeben.

Der Prüfmodus erfolgt jährlich. In den Dienstbesprechungen der Fachgruppe zu Schuljahresbeginn werden die Erfahrungen des vorangehenden Schuljahres gesammelt und bewertet sowie eventuell notwendige Konsequenzen formuliert. Die vorliegende Checkliste wird als Instrument einer solchen Bilanzierung genutzt. Die Ergebnisse dienen dem/der Fachvorsitzenden zur Rückmeldung an die Schulleitung und u.a. an den/die Fortbildungsbeauftragte, außerdem sollen wesentliche Tagesordnungspunkte und Beschlussvorlagen der Fachkonferenz daraus abgeleitet werden. Insgesamt dient die Checkliste über die Evaluation des aktuellen schulinternen Curriculums hinaus zur systematischen Qualitätssicherung und Qualitätsentwicklung der Arbeit der Fachgruppe.

Quelle:

<http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/englisch/hinweise-und-beispiele/schulinterner-lehrplan/schulinterner-lehrplan.htm>



3 Leistungskonzept

3.1 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

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 regelmäßiger Sprechstunden erhalten die Erziehungsberechtigten Gelegenheit, sich über den Leistungsstand ihrer Kinder zu informieren und dabei Perspektiven für die weitere Lernentwicklung zu besprechen.



3.2 Überprüfung der sonstigen 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
 - o Regelmäßige Beteiligung (quantitativ und qualitativ)
 - o Wiedergabe von Kenntnissen und Prozessen
 - o Darstellung von Problemsituationen
 - o Beiträge zur Entwicklung von Problemlösungen und Bewertung von Arbeitsständen und -ergebnissen
 - o Vorstellung von Arbeitsergebnissen aus Gruppen- oder Partnerarbeit
 - o Nutzung adäquater Fachsprache
- Selbstständige Arbeit zu zweit oder in Gruppen.
 - o Mitarbeit bei der Organisation (Absprachen, Arbeitseinteilung, Schnittstellenvereinbarung, gegenseitige Hilfen) der gemeinsamen Arbeit
 - o Kooperatives Verhalten und gemeinsame konstruktive Suche nach Lösungswegen bei den gestellten Problemen
 - o Erreichen des Arbeitszieles unter Einsatz der zur Verfügung stehenden Materialien und der erlernten Arbeitstechniken
- Praktische Arbeit am Computer
 - o Vor- und Nachbereitung der Rechnerarbeitsphasen (u.a. Dokumentationen)
 - o Kooperatives Arbeiten am Rechner und Beratung anderer Arbeitsgruppen
 - o 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
 - o 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 eine 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.

3.3 Überprüfung der schriftlichen Leistung

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

Bei der Bewertung von Klausuren richtet sich die Fachschaft nach den Punktegrenzen des Zentralabiturs (siehe Tabelle).

Die Leistungsbewertung in den Klausuren wird mit Blick auf die schriftliche Abiturprüfung mit Hilfe eines Kriterienrasters („Erwartungshorizont“) durchgeführt, welches neben den inhaltsbezogenen Teilleistungen auch darstellungsbezogene Leistungen ausweist. Den Schülerinnen und Schülern wird auf diese Weise die Leistungsbewertung transparent gemacht.

Die Zuordnung der Hilfspunkte zu den Notenstufen orientiert sich in der Qualifikationsphase am Zuordnungsschema des Zentralabiturs. Die Note ausreichend soll bei Erreichen von 46% der Hilfspunkte erteilt werden. Eine Absenkung der Note kann gemäß APO-GOST bei häufigen Verstößen gegen die Sprachrichtigkeit vorgenommen werden.



Prozent	Punkte	Zen- sur	Prozent	Punkte	Zen- sur	Prozent	Punkte	Zen- sur
100 – 95	15	1+	69 - 65	9	3+	39 - 35	3	5+
94 - 90	14	1	64 – 60	8	3	34 - 30	2	5
89 – 85	13	1-	59 - 55	7	3-	29 - 25	1	5-
84 - 80	12	2+	54 – 50	6	4+	24 - 0	0	6
79 – 75	11	2	49 - 45	5	4			
74 - 70	10	2-	44 - 40	4	4-			

- 39% ist 5+, demnach gibt es bei 40% (also 40 Punkten) eine 4-
- ca. 10% der Gesamtpunktezahl sollten Punkte für die Darstellungsleistung (Ausführung, Struktur, Fachsprache, Form, Gestaltung) sein
- bei der Aufgabenstellung sollten die **Operatoren** verwendet werden
- in den Prüfungen des Zentralabiturs werden die drei **Anforderungsbereiche** in folgender Weise gewichtet:
 - AFB I: 30% (Reproduktion)
 - AFB II: 50% (Transfer, Übertragung von Wissen)
 - AFB III: 20% (Neuvernetzung und selbstständiges Anwenden von Wissen in neuen Problemstellungen, Bewertung)
- die Erwartungshorizonte, die nach der Klausur an die Schüler herausgegeben werden, sollten stark verkürzt sein und lediglich ein paar Stichpunkte bzw. die Punkteverteilung enthalten



3.4 Qualitätssicherung und Evaluation

Die Fachgruppe Informatik bemüht sich um eine stete Sicherung der Qualität ihrer Arbeit. Dazu dient unter anderem die jährliche Evaluation des schulinternen Curriculums mit Hilfe einer Checkliste. Weitere anzustrebende Maßnahmen der Qualitätssicherung und Evaluation sind zum Beispiel gegenseitiges Hospitieren, *team teaching*, Parallelarbeiten oder gemeinsames Korrigieren. Absprachen dazu werden von den in den Jahrgängen parallel arbeitenden Kolleginnen und Kollegen zu Beginn eines jeden Schuljahres getroffen.

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 sich vor allem aus den flexiblen Variablen Schülerzahl, Fachgruppengröße, Lehr- und Lernmittelentwicklung und Abiturvorgaben ergeben.

Der Prüfmodus erfolgt jährlich. In den Dienstbesprechungen der Fachgruppe zu Schuljahresbeginn werden die Erfahrungen des vorangehenden Schuljahres gesammelt und bewertet sowie eventuell notwendige Konsequenzen formuliert. Die vorliegende Checkliste wird als Instrument einer solchen Bilanzierung genutzt. Die Ergebnisse dienen dem/der Fachvorsitzenden zur Rückmeldung an die Schulleitung und u.a. an den/die Fortbildungsbeauftragte, außerdem sollen wesentliche Tagesordnungspunkte und Beschlussvorlagen der Fachkonferenz daraus abgeleitet werden. Insgesamt dient die Checkliste über die Evaluation des aktuellen schulinternen Curriculums hinaus zur systematischen Qualitätssicherung und Qualitätsentwicklung der Arbeit der Fachgruppe.

Quelle:

<http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-ii/gymnasiale-oberstufe/englisch/hinweise-und-beispiele/schulinterner-lehrplan/schulinterner-lehrplan.html>



4 Methodenkonzept

Zusätzlich zu den hier genannten Methoden gilt das allgemeingültige Methodenkonzept des Conrad von Soest Gymnasiums.

Im Fach Informatik finden insbesondere folgende Methoden Anwendung:

Jahrgang 7

- Aufbau eines Computers
- Grundlagen von Windows
- Arbeit mit der Tastatur (Benutzung von ShortCuts)
- Formatierung von Texten
- Tabellarische Textverarbeitung
- Gestalten von Präsentationen unter Verwendung verschiedener Formatierungen und eingebetteter Objekte
- Erstellen zielgruppenorientierter Präsentationen

Jahrgang 8-9

- Text und Bild (mit Hilfe des PC) kombinieren
- Zeitungen gestalten
- Umsetzen von Daten in Diagramme und Graphen
- Internet-Recherche
- Entwerfen von Präsentationsfolien
- Gestaltung und Strukturierung mündlicher Vorträge
- Visuelle Programmierung
- Prozedurale Programmierung
- Grundlagen verschiedener Darstellungsformen von Algorithmen



5 Medienkonzept

siehe allgemeingültiges Medienkonzept des Conrad von Soest Gymnasiums



ANLAGE ZUR LEISTUNGSBEWERTUNG

Bewertungskriterien der mündlichen/praktischen Mitarbeit

Bewertung	Qualität und Quantität der Beiträge	Note
Die Leistung entspricht den Anforderungen in ganz besonderem Maße.	<u>Qualität:</u> sehr gute Kenntnisse über die bisherigen Kursinhalte. Fähigkeit, auch bei komplexen Sachverhalten eigenständig zu problematisieren, zu strukturieren und zusammenzufassen. Sehr gutes Abstraktionsvermögen. Häufiges Einbringen weiterführender Beiträge, auch über außerschulische Fakten und früheren Stoff; verständliche, sichere, flüssige Formulierungen, fehlerfreie und übersichtliche Programmstrukturen. Arbeitet komplett selbstständig, löst effektiv, zügig, sicher und problemorientiert die gestellten Aufgaben, nutzt alle Möglichkeiten Programme kurz zu programmieren. Programme sind strukturiert und fehlerfrei. <u>Quantität:</u> konstante/permanente überragende Mitarbeit während aller Stunden	1
Die Leistung entspricht in vollem Umfang den Anforderungen.	<u>Qualität:</u> gute Kenntnisse über die bisherigen Kursinhalte; Fähigkeit zu strukturieren und zusammenzufassen; gutes Abstraktionsvermögen; Einbringen weiterführender Beiträge, auch über außerschulische Entwicklungen und früheren Stoff; meistens verständliche, flüssige Formulierungen, überwiegend fehlerfrei, übersichtliche Programmstrukturen, überwiegend selbstständig und problemorientiert gelöste Aufgaben, Programmverkürzungen fast auf Minimum <u>Quantität:</u> konstante/ permanente gute Mitarbeit während fast aller Stunden	2
Die Leistung entspricht im Allgemeinen den Anforderungen.	<u>Qualität:</u> zufriedenstellende Kenntnisse über die bisherigen Kursinhalte; Fähigkeit im Rahmen eines teilweise vorgegebenen Lösungsweges zu arbeiten; gelegentliches Einbringen weiterführender Beiträge, auch über außerschulische Entwicklungen und früheren Stoff; verständliche überwiegend sichere Formulierungen. Kann nach entsprechendem Hinweis Programmstrukturen verkürzen und Fehler beheben, erkennt selbst nur selten den kürzesten Programmierweg, <u>Quantität:</u> grundsätzliche Mitarbeit in allen Stunden	3
Die Leistung weist zwar Mängel auf, entspricht im Ganzen aber noch den Anforderungen.	<u>Qualität:</u> teilweise lückenhafte Kenntnisse über die bisherigen Kursinhalte; kann in einer vorgegebenen Struktur arbeiten; wenige Beiträge, oft reproduktiv aus abgegrenztem Gebiet in gelerntem Zusammenhang; verständliche, aber knappe, kurze Formulierungen, u. U. in unvollständigen Sätzen angemessene aber teilweise	4



	<p>fehlerhafte Auseinandersetzung mit geforderter Software/Programmierungsumgebung, kommt mit Hilfestellung zu Teillösungen, hat Probleme Teillösungen zusammenzufügen. Äußerliche Programmstrukturen werden noch kaum eingehalten. Oft umständliche Programmierungen – zu lang.</p> <p><u>Quantität:</u> unregelmäßige Mitarbeit, nicht in allen Stunden; oft nur nach Aufforderung</p>	
<p>Die Leistung entspricht den Anforderungen nicht, notwendige Grundkenntnisse sind jedoch vorhanden und die Mängel in absehbarer Zeit behebbar.</p>	<p><u>Qualität:</u> stark lückenhafte Kenntnisse; ist auch unter Anleitung nicht fähig, Beiträge zu strukturieren; kaum Beiträge, wenn, dann meist als unstrukturierte Teilergebnisse; häufig unpräzise Formulierungen, kaum aktive Auseinandersetzung mit geforderter Software, unter Anleitung kaum fähig Aufgaben am Rechner zu bewältigen, kann maximal kleine Teilergebnisse am Rechner liefern.</p> <p><u>Quantität:</u> gelegentliche, äußerst seltene Mitarbeit, nur nach Aufforderung</p>	5
<p>Die Leistung entspricht den Anforderungen nicht. Selbst Grundkenntnisse sind so lückenhaft, dass die Mängel in absehbarer Zeit nicht behebbar sind.</p>	<p><u>Qualität:</u> minimale Kenntnisse; keine Beiträge, auch nicht auf Nachfragen, keine aktive Auseinandersetzung mit geforderter Software, fast ausschließlich keine oder themenfremde Beschäftigung mit dem Rechner, keine Anstrengungsbereitschaft nach Aufforderung und Hilfestellung</p> <p><u>Quantität:</u> keine Mitarbeit</p>	6

