

Schulinternes Curriculum

für das Fach

Informatik

am

Städtischen Gymnasium

Schmallenberg

Stand: 01/2012

Inhaltsverzeichnis

Teil A: Allgemeine Grundlagen S I / S II	Seite 4 - 9
1. Vorbemerkung	4
2. Die Bedeutung des Unterrichtsfachs Informatik	4
3. Die Arbeitsweisen	5
3.1 Lehrende	5
3.2 Lernende	5
4. Fachübergreifende und fächerverbindende Sichtweisen.....	6
5. Leistungsfeststellung und Leistungsbewertung	6
5.1 Grundsätze.....	6
5.1.1 Umgang mit Lernerfolgskontrollen.....	6
5.1.2 Bewertung der Produkte sowie des Prozesses.....	6
5.1.3 Bereiche, in denen Leistung gefordert wird	7
5.2 Bereiche der Bewertung	7
5.2.1 Das Unterrichtsgespräch.....	7
5.2.2 Projektarbeit	7
5.2.3 Prozessdokumentation	7
5.2.4 Produkte	8
5.2.5 Vorträge von Schülerinnen und Schülern	8
5.2.6 Gespräche zur Überprüfung des Lernerfolges.....	9
5.2.7 Schriftliche Lernerfolgskontrollen	9
5.2.8 Klassenarbeiten und Klausuren	9
5.3 Definitionen der Notenstufen.....	9
5.3.1 Bereich „sonstige Mitarbeit“	9
5.3.2 Bereich Klassenarbeiten/Klausuren.....	10
5.4 Abschlussnote	11
6. Individuelle Förderung.....	11
Teil B: Unterrichtspraktischer Teil S I (Differenzierung II Klasse 8/9)	Seite 12 - 14
Teil C: Allgemeine Grundlagen S II als Erweiterung zu Teil A (S. 4 – 9)	Seite 14 – 16
1. Vorbemerkung	14
2. Lernziele des Schullehrplans.....	14
2.1 Lernziele Informatik NRW.....	14
2.2 Vorgaben zum Zentralabitur.....	15
2.2.1 Inhaltliche Schwerpunkte	15
2.2.2 Lern- und Prüfungsbereiche	15
3. Die besondere Funktion der Einführungsphase	16
Teil D: Unterrichtspraktischer Teil S II	Seite 16 – 25
1. Einführungsphase	16
1.1 Grundlagen der Programmierung.....	16
1.2 Unterrichtsinhalte.....	17
1.2.1 Verbindliche Unterrichtsinhalte	17
1.2.2 Fakultative Unterrichtsinhalte	18
1.3 Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen	18
2. Q1.....	18
2.1 Q1/1: Konzepte und Anwendungen der Theoretischen Informatik.....	18
2.1.1 Begründung.....	18
2.1.2 Unterrichtsinhalte.....	20

2.1.2.1 Verbindliche Unterrichtsinhalte: Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen.....	20
2.1.2.2 Fakultative Unterrichtsinhalte: Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen.....	21
2.2 Q1/2: Objektorientierte Modellierung und Datenstrukturen	21
2.2.1 Unterrichtsinhalte.....	21
2.2.1.1 Verbindliche Schwerpunkte: Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen.....	21
2.2.2 Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen	22
3. Q2.....	22
3.1 Beispiele.....	22
3.1.1 Q2-Wahlthema: Computergraphik	22
3.1.1.1 Begründung.....	22
3.1.1.2 Unterrichtsinhalte.....	23
3.1.2 Q2-Wahlthema: Simulationen	23
3.1.2.1 Begründung.....	23
3.1.2.2 Unterrichtsinhalte.....	23
3.1.3 Q2-Wahlthema: Rechnerstrukturen	24
3.1.3.1 Begründung.....	24
3.1.3.2 Unterrichtsinhalte.....	24
3.1.4 Q2-Wahlthema: Rechnernetze	24
3.1.4.1 Begründung.....	24
3.1.4.2 Unterrichtsinhalte/Aufgaben.....	25

Teil A: Allgemeine Grundlagen S I / S II

1. Vorbemerkung

Bei der Erstellung und Umsetzung eines schulinternen Curriculums für das Fach Informatik ergeben sich folgende Probleme:

1. Der Lehrplan für die Sekundarstufe 1 stammt aus dem Jahr 1993 und ist damit veraltet. Als er verfasst wurde, kannte kaum jemand das Internet; Windows 95 war noch nicht entwickelt; und Vieles, was heute selbstverständliches Wissen jedes Grundschülers ist (z. B. „Wie speichere ich eine Datei?“) war damals noch vielen Schülerinnen und Schülern unbekannt. Auch die Richtlinien und Lehrpläne für die Sekundarstufe II NRW sind in Teilen veraltet (1. Auflage 1999). Eine Bezugnahme auf diese Lehrpläne ist, zumindest soweit es konkrete Unterrichtsinhalte angeht, daher schwierig. Darüber hinaus fehlt die Kompetenz-Orientierung, die alle neueren Lehrpläne aufweisen.
2. Es gibt für das Fach Informatik in NRW keine für den Lehrplan zugelassenen Lehrwerke. Auch ein Lehrwerk kann deshalb nicht auf dieselbe Weise wie in anderen Fächern die Grundlage für das schulinterne Curriculum bilden.
3. Nicht nur die Inhalte, die auch in anderen Fächern Änderungen unterliegen, sondern auch die Methoden der Informatik wandeln sich mit großer Geschwindigkeit. Auch die Lebenswelt der Schülerinnen und Schüler wandelt sich, hinsichtlich ihres Umgangs mit Computern und neuen Informationsformen, in beinahe jährlichem Rhythmus. Je konkreter Inhalte und Methoden deshalb im Curriculum festgelegt werden, desto schneller veraltet dieses Curriculum auch.
4. Die Umsetzbarkeit eines pädagogisch-didaktischen Konzeptes entscheidet letztendlich zu erheblichen Teilen über den Erfolg desselben und damit über den Qualitätsstandard eines Faches. Im Fach Informatik setzen dabei in besonderer Weise die materialen und technisch-instrumentellen Bedingungen entscheidende Parameter für eine erfolgreiche Umsetzung eines solchen Konzeptes. Die Ausstattung des Fachraumes Informatik sollte zielgenau auf die Bedürfnisse des Fachunterrichtes abgestellt sein. Die Formulierung dieser Bedürfnisse ist wesentlicher Bestandteil des Aufgabenkataloges der Fachschaft Informatik. Die Kompetenzen von Fachlehrern verdienen deshalb in diesem Zusammenhang unbedingte Beachtung.

Die konsequente Anwendung dieser Grundsätze bei der Implementierung der Informatik am Städtischen Gymnasium Schmallenberg als reguläres Unterrichtsfach Anfang der 1990er Jahre begründete in den Folgejahren den hohen Stellenwert u. a. bei der Ausbildung eines Schulprofils und nicht zuletzt als gefragtes Abiturfach. Vor diesem Hintergrund formulierte im Jahr 2006 die Fachkonferenz anlässlich der Neuausstattung des Fachraumes den aktuellen Bedarf. Es bleibt festzustellen, dass bei der Beschaffung eine Mitwirkung der Fachlehrer und die Nutzung ihrer Kompetenzen seitens der Entscheidungsträger unterblieb. Daraus resultierend ergeben sich leider auch negative Auswirkungen bis in die tägliche Unterrichtspraxis (lange Startzeiten, Geschwindigkeit des Datenverkehrs, Störanfälligkeit der pädagogischen Betreuungssoftware, mangelnde Wartung, u.a.m.), die die Qualität und den Erfolg des Unterrichts beeinträchtigen.

Das vorliegende Curriculum verweist teilweise auf die in Hessen eingeführten Richtlinien.

2. Die Bedeutung des Unterrichtsfachs Informatik

In der Informations- und Wissensgesellschaft spielen komplexe Informatiksysteme eine wachsende Rolle im täglichen Leben und verändern in zunehmendem Maße die Arbeits- und Lebensweise der Menschen. Die Informatik als Bezugswissenschaft nimmt in diesem gesellschaftlichen Wandel eine Schlüsselrolle ein, da sie sich mit den theoretischen Grundlagen, Methoden und Verfahren der Informations- und Kommunikationstechniken beschäftigt, welche auch gegenüber der ungemein schnellen technischen Entwicklung Bestand haben.

Das Schulfach Informatik ergänzt die für eine zeitgemäße Allgemeinbildung erforderliche Betrachtung der Welt um informatische Aspekte und leistet damit neben den naturwissenschaftlichen Fächern und der Mathematik einen spezifischen, eigenständigen Beitrag im mathematisch-naturwissenschaftlich-technischen Aufgabenfeld.

Die Methoden und Werkzeuge der Informatik, die Denk- und Herangehensweisen sowie die Nutzung von Informatiksystemen findet inzwischen in fast allen Gebieten von Wissenschaft, Wirtschaft und Technik Eingang – die Geistes- und Gesellschaftswissenschaften eingeschlossen.

3. Die Arbeitsweisen

3.1 Lehrende

Im Fach Informatik bieten sich in besonderer Weise kooperative Arbeitsweisen auch für die Lehrer an. Am Städtischen Gymnasium Schmalleberg wird seit mehr als zehn Jahren die Planung und Durchführung von Unterrichtsvorhaben in enger kollegialer Abstimmung praktiziert, die Synergieeffekte zum Nutzen der Kollegen und auch der Schüler freisetzt. Diese enge Abstimmung findet ebenfalls Anwendung für die Formulierung von Leistungsanforderungen und Lernerfolgskontrollen. So ist es die Regel, in parallelen Lerngruppen gemeinsame Klassenarbeiten und Klausuren zu schreiben und nach einem einheitlichen Bewertungsmodus zu bewerten. Daraus resultiert eine weitgehende Vergleichbarkeit und Transparenz im Sinne einer Standardisierung.

3.2 Lernende

Ohne grundlegende informatikspezifische Methoden lassen sich Modelle zu gegebenen Aufgabenstellungen kaum entwickeln. Da diese in der Regel zu komplex sind, um in einem einzigen Arbeitsgang erfasst oder gar gelöst zu werden, ist als erster Schritt das Problem in einfachere Teilprobleme zu zerlegen und bis auf die zu beschreibenden Schnittstellen in voneinander unabhängige Teilaufgaben aufzuteilen (Prinzip der Modularisierung).

Der nächste Schritt ist dann die Wahl einer geeigneten Datenorganisation in der verwendeten Programmierumgebung. Diese gibt ihrerseits die Werkzeuge vor, mit denen eine algorithmische Lösung der einzelnen Module erarbeitet und realisiert werden kann. Dabei wird man sich zunächst auf die in der Programmiersprache vorgegebenen Möglichkeiten beschränken und ihre Funktionsweisen aufdecken, um sie im Rahmen der Aufgabenlösung sinnvoll einsetzen zu können.

Mit fortgeschrittener Erfahrung in der Lerngruppe wird man dazu übergehen, aus den gegebenen Standards der Sprache kreativ neue und leistungsfähigere Datentypen und Werkzeuge zu entwickeln – im klassischen Sinne wären das allgemeine (abstrakte) Datentypen. Methodisch wird dies auf natürliche Weise durch komplexere Aufgabenstellungen erreicht, die mit elementaren Objekten und Werkzeugen nicht problemadäquat zu bewältigen wären. Die weiteren methodischen Schritte sind die Analyse und kritische Bewertung der Problemlösung und der Funktionalität des Programms mit ggf. einer entsprechenden Modifizierung.

Die aktive Auseinandersetzung mit den Inhalten und Methoden der Informatik im Unterricht erfordert eine intensive Nutzung der schulischen Computeranlagen. Über das Internet können Arbeitsergebnisse zwischen Schule und häuslichem Arbeitsplatz ausgetauscht werden. Die Interaktivität des Mediums Computer ermöglicht in besonderer Weise selbstständiges und selbsttätiges Arbeiten und Lernen der Schülerinnen und Schüler. Damit sind günstige Voraussetzungen zum Lernen nach gemäßigt konstruktivistischer Sicht gegeben, nach welcher Lernen als aktiv-konstruktiver, selbstgesteuerter, situativer und sozialer Prozess verstanden wird. Als Leitkonzept für die Gestaltung geeigneter Lernumgebungen eignet sich die Problemorientierung, die eine Balance zwischen Instruktion und Konstruktion einfordert. Richtungsweisende Gestaltungsprinzipien sind authentische, multiple, soziale und instruktionale Kontexte. Die Themen entstammen daher möglichst realen Problemen mit Anwendungsbezug, die mehrere Sichtweisen zu einem Inhalt erlauben.

Soziale Kontexte werden durch Gruppen und Teamarbeit geschaffen. Wo Anleitung und Unterstützung erforderlich ist, muss sie den Lernenden gewährt werden. Problemorientierung im hier beschriebenen Sinn lässt sich sinnvoll mit themen- und projektorientierten wie auch in fächerverbindenden oder fachübergreifenden Unterrichtsphasen umsetzen.

Zusammengefasst soll sich die Unterrichtsorganisation daran ausrichten, dass

- die individuelle Schülerpersönlichkeit mit ihren Vorerfahrungen, Möglichkeiten und Leistungsdispositionen im Blick ist,
- Schülerinnen und Schüler aktiv lernen,
- Schülerinnen und Schüler kooperativ lernen,
- Vorwissen abgesichert, aufgegriffen und Lernfortschritt ermöglicht wird,

- die Aufgabenstellungen komplex sind,
- die Aufgabenstellungen auch auf Anwendung und Transfer ausgerichtet sind.

4. Fachübergreifende und fächerverbindende Sichtweisen

Der Anfangsunterricht in Informatik ist durch fachspezifische Gegenstände geprägt wie das Erlernen von Elementen einer geeigneten Sprache und die Einarbeitung in eine Programmierumgebung, in der sich Lösungen angemessen formulieren lassen. Dabei stellt sich bald heraus, dass die Aufgaben, für die man algorithmische Lösungen sucht, aus ganz unterschiedlichen Fachbereichen stammen. Insofern fordert die Informatik eine *fächerverbindende und fachübergreifende Betrachtungsweise* geradezu heraus. Grundsätzliche Gedanken hierzu sind bereits in Kapitel 3.2 dargestellt.

Durch eine Vielzahl von Algorithmen hat die Informatik inzwischen erhebliche Dienstleistungen für andere Fächer erbracht. Hierzu gehören umfangreiche mathematische Berechnungen, Simulationen von Zufallsprozessen und dynamischen Systemen, Verfahren der Linguistik, Verfügbarmachung „künstlicher Intelligenz“ und die Bereitstellung einer immer komplexer organisierten Informations- und Kommunikationstechnologie für vernetzte Systeme. Der Bezug zu anderen Fachbereichen und zu fächerverbindenden Themen ist also unmittelbar vorhanden und muss nicht erst künstlich hergestellt werden.

Angesichts der Verzahnung mit anderen Bereichen erscheint es im Gegenteil eher notwendig, im Laufe des Unterrichts das spezifisch Informatische als eigenständigen fachlichen Kern herauszuarbeiten. Auf die Algorithmik bezogen bedeutet das etwa, den Beitrag der Informatik zur Entwicklung von Strategien des Denkens hervorzuheben und diese zu bewerten.

Zusammenfassend lässt sich feststellen: Der fachübergreifende Aspekt der Informatik liegt in ihren Anwendungsfeldern. Der fächerverbindende Aspekt ergibt sich durch die allgemeinen Methoden und Systeme, die bei Problemlösungen in verschiedenen Fachbereichen benutzt werden.

5. Leistungsfeststellung und Leistungsbewertung

5.1 Grundsätze

Grundlage der Leistungsbewertung sind § 6 der Verordnung über die Ausbildung und die Abschlussprüfungen in der Sekundarstufe I (APO-SI) sowie §§ 13 - 16 der Abitur- und Prüfungsordnung der Gymnasialen Oberstufe (APO-GOST). Weitere Grundlagen der Bewertung stammen aus zwei Bereichen: einerseits aus den Beobachtungen des Lernprozesses, andererseits aus den mündlichen und schriftlichen Lernerfolgskontrollen.

Der Lernprozess wird charakterisiert durch die Lernbereitschaft, das Lernverhalten, die Fähigkeit, das eigene Lernen zu beobachten und aus Fehlern zu lernen, sowie die Fähigkeit zum Lernen durch Wechselwirkung mit der Lerngruppe.

5.1.1 Umgang mit Lernerfolgskontrollen

Die Leistungsbewertung setzt voraus, dass den Schülerinnen und Schülern die inhaltlichen und methodischen Anforderungen jeder Unterrichtssequenz klar sind. Es muss ihnen genügend Gelegenheit zur Übung gegeben werden. Die Kriterien der Beurteilung müssen für die Schülerinnen und Schüler transparent sein. Nur so fördern sie deren Fähigkeit zur Selbsteinschätzung und tragen dazu bei, dass sie ihren eigenen Lernprozess bewusst wahrnehmen und beurteilen können. Daher werden den Schülerinnen und Schülern die Bereiche der Bewertung (s. u.) zu Beginn des Schuljahres bekannt gegeben.

5.1.2 Bewertung der Produkte sowie des Prozesses

Lernerfolgsüberprüfungen sind ein kontinuierlicher Prozess. Besonders die fachtypischen offenen Lernsituationen erfordern die begleitende Beobachtung der Entwicklung von Schülerleistungen. Zur Beurteilung der Schülerleistung kann nicht nur ein fertiges Produkt herangezogen werden, sondern es müssen auch die Ausgangslage und Zwischenschritte berücksichtigt werden. Deshalb werden neben den Produkten außerdem die dazugehörigen Prozessdokumentationen bewertet. Zum Erbringen der geforderten Leistungen muss den Schülerinnen und Schülern genügend Zeit gegeben werden.

5.1.3 Bereiche, in denen Leistung gefordert wird

Bewertungen sollten unter Einbeziehung aller Kompetenzbereiche sowohl produkt- als auch prozessorientiert erfolgen.

Sie umfassen

- die formale und inhaltliche Qualität der erstellten Produkte,
- das Niveau der Reflexion (Begründung, Erläuterung) der eigenen Arbeit,
- das Niveau der Darstellung und Beurteilung von Sachverhalten,
- die Selbstständigkeit der Arbeit (Aneignung, Produkterstellung),
- die Kommunikationskompetenz und Teamfähigkeit,
- die Sicherheit und Flexibilität im Umgang mit den vermittelten Werkzeugen und Methoden.

5.2 Bereiche der Bewertung

5.2.1 Das Unterrichtsgespräch

Unterrichtsgespräche sind Gesprächssituationen in der gesamten Lerngruppe. Die Gesprächsbeiträge der Schülerinnen und Schüler werden nach folgenden Aspekten beurteilt:

- situationsgerechte Einhaltung der Gesprächsregeln
- Anknüpfung an Vorerfahrungen und den erreichten Sachstand
- sachliche, begriffliche und sprachliche Korrektheit
- Verständnis anderer Gesprächsteilnehmer und Bezug zu ihren Beiträgen
- Ziel- und Ergebnisorientierung
- Eigenständigkeit und Verständlichkeit
- Verwendung der informatischen Fachsprache
- Knüpfen logischer Zusammenhänge
- Anpassung bekannter Methoden zur Lösung neuartiger Probleme
- Reflexionskompetenz

5.2.2 Projektarbeit

Im Informatikunterricht haben projektorientierte Arbeitsformen einen bedeutenden Stellenwert.

Beurteilt werden als **Individuelleistung**:

- Anspruchsniveau der Aufgabenauswahl
- Beachtung der Aufgabenstellung
- Einhaltung verbindlicher Absprachen und Regeln
- konzentriertes, zügiges und verantwortungsbewusstes Arbeiten
- Aufgeschlossenheit und Selbstständigkeit, Lösungen für Probleme zu finden
- Übernahme der Verantwortung für den eigenen Aufgabenbereich
- Einsatz und Erfolg bei der Informationsbeschaffung
- Flexibilität und Sicherheit im Umgang mit den Werkzeugen.

Die **Leistung im Team** wird beurteilt nach:

- Voranbringen der Gruppenarbeit durch eigene Initiative
- Strukturierung der Gruppenarbeit
- Lösen der eigenen Teilaufgabe und Abstimmung mit den Anderen
- Einbringen und Vertreten eigener Ideen
- Nachvollziehen und Einordnen von Ideen anderer Gruppenmitglieder
- Weiterentwickeln von Vorschlägen anderer Gruppenmitglieder
- Konstruktives Aufnehmen von Kritik an eigenen Vorschlägen

5.2.3 Prozessdokumentation

Die Prozessdokumentation enthält für jeden Arbeitsabschnitt Beschreibungen zur individuellen Ausgangslage, zur eigenen Teilaufgabe, zur Vorgehensweise, zu den aktuellen Tätigkeiten und Ergebnissen sowie zu den

Lernfortschritten. Hier wird der Lernprozess dokumentiert, wobei deutlich wird, wie die Schülerin oder der Schüler mit Irrwegen und Fehlern umgeht.

Beurteilt werden:

- Umfang und Strukturierung der Darstellung
- Übersichtlichkeit und Sorgfalt
- sachliche Korrektheit
- Verwendung der informatischen Fachsprache
- Informationsdichte
- Fähigkeit, Neues zu erkennen, einzuordnen und zu bewerten
- konstruktiver Umgang mit Fehlern
- Arbeitsbereitschaft
- Lernbereitschaft

5.2.4 Produkte

Produkte sind beispielsweise Darstellungen von Modellierungen, Informatiksysteme oder Teile davon und die dazugehörigen Dokumentationen und/oder Präsentationen.

Beurteilt werden:

- inhaltliche Bewältigung der Aufgabe
- Verständnis für die fachbezogenen Methoden
- Dokumentation des Lösungsweges
- sachliche Korrektheit
- Schwierigkeitsgrad
- Folgerichtigkeit
- Originalität
- Adressatenbezug
- sachangemessene sprachliche Darstellung unter Verwendung der Fachsprache
- Umfang und Vielfalt der fachbezogenen Aspekte
- Sorgfalt und optische Umsetzung

5.2.5 Vorträge von Schülerinnen und Schülern

Die Schülerinnen und Schüler können einzeln oder in einer kleinen Gruppe ihre Arbeitsergebnisse oder ein selbst erarbeitetes Themengebiet präsentieren.

Beurteilt werden:

- inhaltliche Bewältigung der Aufgabe
- Zuhörerorientierung
- Verständnis für informatische Methoden
- sachliche Korrektheit
- optische oder akustische Aufbereitung
- Schwierigkeitsgrad
- Folgerichtigkeit
- Originalität
- sachangemessene sprachliche Darstellung unter Verwendung der Fachsprache
- Umfang und Vielfalt der fachbezogenen Aspekte
- Auftreten und Vortragsstil
- Ertragen und Aufnehmen von Kritik
- Fähigkeit, situationsangemessen auf Fragen zu reagieren
- Fähigkeit, als Zuhörer Fragen zu stellen und Kritik zu formulieren

5.2.6 Gespräche zur Überprüfung des Lernerfolges

Gespräche zwischen Lehrenden und Lernenden während der laufenden Arbeit helfen, Vorgehensweise und Fortschritte zu reflektieren und Fehlentwicklungen zu vermeiden. Gespräche während des Lernprozesses sind anders zu bewerten als Gespräche zur Leistungsüberprüfung.

Beurteilt werden:

- Analyse und Strukturierung der Problemstellung
- informatische Sach- und Methodenkenntnisse
- Beschaffen und Einbringen von Informationen
- kritische Auseinandersetzung mit Informationen
- Knüpfen logischer Zusammenhänge
- Entwickeln und Anwenden von Modellvorstellungen
- Entwerfen von Lösungswegen
- Problemlösen mithilfe des Computers
- Strategien bei der Fehlersuche
- Verwendung der informatischen Fachsprache

5.2.7 Schriftliche Lernerfolgskontrollen

Schriftliche Lernerfolgskontrollen sind Hausarbeiten, Protokolle, Tests, Klassenarbeiten, Klausuren.

Beurteilt werden:

- sachliche, begriffliche und sprachliche Korrektheit
- Verwendung der informatischen Fachsprache
- Übersichtlichkeit, Lesbarkeit und Verständlichkeit
- Eigenständigkeit und Originalität der Bearbeitung und Darstellung
- Anpassung bekannter Methoden zur Lösung neuartiger Probleme
- Reichhaltigkeit und Vollständigkeit
- Nachvollziehbarkeit der bei Problemlösungen getroffenen Entscheidungen
- Reflexionskompetenz

5.2.8 Klassenarbeiten und Klausuren

Je Schulhalbjahr der Sekundarstufe I werden zwei Klassenarbeiten im Umfang von 1 - 2 Unterrichtsstunden à 45 Minuten geschrieben. Im Schuljahr kann eine Klassenarbeit durch die Bearbeitung eines größeren Projektes ersetzt werden (vgl. § 6(8) und VV 6.1.1 APO-S I).

Die Zahl und die Dauer der Kursarbeiten in der S II richtet sich nach § 14 APO-GOST

Bei der Beurteilung von Klassen- und Kursarbeiten ist darauf zu achten, dass nicht nur die Richtigkeit der Ergebnisse und die inhaltliche Qualität, sondern auch die angemessene Form der Darstellung ein wichtiges Kriterium der geforderten Leistung ist.

5.3 Definitionen der Notenstufen

5.3.1 Bereich „sonstige Mitarbeit“

Die Note „**sehr gut**“ wird erteilt, wenn folgende Kriterien überwiegend zutreffen:

- die Beteiligung ist regelmäßig und erfolgt meist in Eigeninitiative
- dem Unterricht wird stets konzentriert gefolgt
- die Beiträge sind sachbezogen, fachlich richtig und hochwertig
- fachbezogene, kritisch-konstruktive und weiterführende Fragen werden gestellt
- die Beiträge führen den Unterricht weiter
- Gelerntes wird auf neue Sachverhalte übertragen
- es wird vollständig über das erarbeitete Unterrichtswissen verfügt
- alle Anforderungen werden regelmäßig, eigenständig und vollständig erfüllt
- die Listings sind strukturiert und weitgehend kommentiert
- die Ergebnisse können für alle Schüler als Musterlösung vervielfältigt werden

Die Note „**gut**“ wird erteilt, wenn folgende Kriterien überwiegend zutreffen:

- die Beteiligung erfolgt häufig und meist in Eigeninitiative
- dem Unterricht wird überwiegend konzentriert gefolgt
- die Beiträge sind fachlich richtig und fachsprachlich korrekt formuliert
- Ergebnisse können zusammengefasst und begründet werden
- die Beiträge führen den Unterricht weiter
- Gelerntes kann auf neue Sachverhalte übertragen werden
- es wird vollständig über das erarbeitete Unterrichtswissen verfügt

Die Note „**befriedigend**“ wird erteilt, wenn folgende Kriterien überwiegend zutreffen:

- die Beteiligung erfolgt regelmäßig und meist in Eigeninitiative
- die Beiträge sind im Wesentlichen fachlich richtig
- dem Unterricht wird regelmäßig gefolgt
- es wird über sichere Grundkenntnisse verfügt
- erlernte Verfahren können angewendet werden

Die Note „**ausreichend**“ wird erteilt, wenn folgende Kriterien überwiegend zutreffen:

- die Beteiligung erfolgt unregelmäßig
- die Beteiligung erfolgt oft nur nach Aufforderung
- die Beiträge sind eher knapp
- erlernte Verfahren können - teilweise mit kleineren Fehlern – angewendet werden

Die Note „**mangelhaft**“ wird erteilt, wenn folgende Kriterien überwiegend zutreffen:

- die Beteiligung erfolgt selten
- die Beteiligung nach Aufforderung zeigt oftmals fachliche Mängel
- in sach- und fachbezogenen Diskussionen zeigt sich ein passives Verhalten
- es wird über Grundkenntnisse verfügt
- die erlernten Verfahren werden nicht sicher beherrscht

Die Note „**ungenügend**“ wird erteilt, wenn folgende Kriterien überwiegend zutreffen:

- keine Beteiligung am Unterrichtsgeschehen
- bei Aufforderung können Fragen nicht korrekt beantwortet werden
- es wird nicht über gesicherte Grundkenntnisse verfügt

5.3.2 Bereich Klassenarbeiten/Klausuren

Die Leistung – speziell in der Sekundarstufe II hinsichtlich der Abiturprüfung - wird in drei Anforderungsbereichen bewertet:

Der **Anforderungsbereich I** umfasst

- die Wiedergabe von bekannten Sachverhalten aus einem abgegrenzten Gebiet im gelernten Zusammenhang.
- die Beschreibung und Darstellung bekannter Verfahren, Methoden und Prinzipien der Informatik.
- die Beschreibung und Verwendung gelernter und geübter Arbeitstechniken und Verfahrensweisen in einem begrenzten Gebiet und in einem wiederholenden Zusammenhang.

Der **Anforderungsbereich II** umfasst

- die selbstständige Verwendung (Auswählen, Anordnen, Verarbeiten und Darstellen) bekannter Sachverhalte zur Bearbeitung neuer Frage- oder Problemstellungen unter vorgegebenen Gesichtspunkten in einem durch Übung bekannten Zusammenhang.
- die selbstständige Übertragung des Gelernten auf vergleichbare neue Situationen, wobei es entweder um veränderte Fragestellungen oder um veränderte Sachzusammenhänge oder um abgewandelte Verfahrensweisen gehen kann.

Der **Anforderungsbereich III** umfasst

- die planmäßige Verarbeitung komplexer Gegebenheiten mit dem Ziel, zu selbstständigen Gestaltungen bzw. Deutungen, Folgerungen, Begründungen, Wertungen zu gelangen.

- die bewusste und selbstständige Auswahl und Anpassung geeigneter gelernter Methoden und Verfahren in neuartigen Situationen, wobei aus diesen die zur Bewältigung der Aufgabe geeigneten selbstständig ausgewählt und einer neuen Problemstellung angepasst werden.

Sowohl in der Beurteilung von Schülerleistungen in der Sekundarstufe I als auch in der Sekundarstufe II werden die Anforderungsbereiche zur Beschreibung der Qualität der Leistung genutzt.

Als Grundlage der Leistungsbewertung erfolgt eine Bepunktung der einzelnen Aufgaben anhand des jeweiligen Schwierigkeitsgrades der (Teil-)Aufgaben. Dabei gilt, dass zum Erzielen einer mindestens ausreichenden Leistung die erfolgreiche Lösung aller Aufgaben des Anforderungsbereichs I ausreichend sind und zur Erzielung einer guten Lösung starke Anteile an richtig gelösten Aufgaben der Anforderungsbereiche II und III notwendig sind. Sehr gute Leistungen sollen nur dann erreicht werden können, wenn insbesondere komplexere Transferaufgaben selbstständig und umfassend gelöst worden sind. Von der Gesamtpunktzahl sollen dabei für eine ausreichende Leistung in der Regel mindestens 40 %, für eine gute Leistung mindestens 70 % erreicht worden sein.

5.4 Abschlussnote

Grundlage der Abschlussnote im Schulhalbjahr sind alle von der Schülerin oder dem Schüler im Beurteilungsbereich „Klassenarbeiten“ und im Beurteilungsbereich „Sonstige Leistungen im Unterricht“ erbrachten Leistungen. Beide Beurteilungsbereiche sind mit vergleichbarem Stellenwert zu berücksichtigen (vgl. § 6(3) APO-S I und § 13(1) APO-GOST).

Die Gesamtnote setzt sich in der Sekundarstufe I bzw. der Sekundarstufe II (bei Wahl als schriftliches Fach) aus den beiden Quartalsnoten der sonstigen Mitarbeit und den Noten der beiden schriftlichen Arbeiten zusammen (50 : 50). Da in der Einführungsphase nur jeweils eine Klausur geschrieben wird, ist deren Anteil an der Gesamtnote dort nicht mehr als 40%.

6. Individuelle Förderung

Insbesondere während der Rechnerarbeitsphasen finden sich im Fach Informatik vielschichtige Differenzierungsmöglichkeiten. So kann bei Projekten die Aufteilung gemäß der einzelnen Lernfortschritte gewählt werden, so dass jeder Schüler seinen sinnvollen Beitrag zum Gesamtergebnis beitragen kann und seine Fähigkeiten im Rahmen des regen Austausches in und zwischen den Gruppen ausbauen kann.

Durch den stark problemorientierten Aufbau der meisten Arbeitsprojekte/Sequenzen erhalten die Schüler die Möglichkeit, sich rasch auch in fremde Themengebiete einzuarbeiten. Dabei wird den unterschiedlichen Lernertypen die vielfältige Auswahl an Themen gerecht. Schüler, deren Denkprozesse sich eher an mathematisch-logisch Prozessen orientieren, finden in denjenigen Einheiten, die algorithmische Beschreibungen von Vorgängen in den Vordergrund stellen, sowie in der Reihe der Theoretischen Informatik (in Q1.1 und Q2), eine sinnvolle Unterstützung; diejenigen, deren Auffassung sich an praktisch-konkreten Zielen orientiert, werden sehr erfolgreich in modellierenden und simulierenden Sequenzen arbeiten können. Dadurch, dass beide Vorgehensweisen in Anteilen in allen informatischen Projekten Verwendung finden, ist sicher gestellt, dass unterschiedliche Lernertypen stets die Möglichkeit zur erfolgreichen Teilnahme am Unterricht und zu einem ausgeprägten persönlichen Lernzuwachs erhalten.

Großer Wert wird in allen Halbjahren darauf gelegt, dass die praktische Umsetzung der theoretischen Grundlagen am Ende jeder Sequenz in einer Form erfolgt, bei der die Schüler möglichst selbstständig die Planung, Koordination und Durchführung der Projekte vornehmen. Auch hier sind deutliche individuelle Lernerfolge durch die starke Aufgliederung der Arbeitsanforderungen zu erwarten.

Teil B: Unterrichtspraktischer Teil S I (Differenzierung II Klasse 8/9)

Das Fach Informatik wird am Städtischen Gymnasium Schmalleberg im Rahmen der Differenzierung II in Form eines 3-stündigen Kurses angeboten.

Die im Fach Informatik aktuell für die Sekundarstufe I gültigen verbindlichen Richtlinien und Lehrpläne des Landes NRW stammen aus dem Jahr 1993. Die Auswahl der Unterrichtsinhalte, Methoden und die Leistungsbewertung orientieren sich daher unter anderem an den von der Gesellschaft für Informatik (GI) herausgegebenen Bildungsstandards.

Die Auswahl der inhaltlichen Kompetenzen legt ihren Schwerpunkt auf folgende Säulen:

- Reflexion über und Analyse von bereits fertigen Produkten und Programmen: Medienkompetenz
- Einblick in den Computer und seine Arbeitsweise → Technische Informatik
- Anwendung von Software und Handhabung des Rechners als Arbeitswerkzeug
- Gestaltung und Entwicklung eigener Produkte und Programme

Weitere grundlegende Kompetenzen, die das Fach Informatik vermitteln möchte, sind sprachliche Ausdrucksfähigkeit, Umgang mit mathematischen Systemen, Verfahren und Modellen. Ebenso sollen erste Einblicke in das Denken und Arbeiten mit komplexen Systemen gegeben werden.

Die genauen Inhalte der einzelnen Schulhalbjahre begründet und verstärkt durch die Beschreibung von Zielen, Kompetenzen und Methoden lauten:

Anhand der Inhalte für das Halbjahr 8.1 werden die Bereiche „Anwendersysteme“, „Informations- und Kommunikationssysteme“ (z. B. MS Office) bearbeitet. Die Reihenfolge der Inhalte (graphische Benutzeroberfläche, Dateiverwaltung, Textverarbeitung, Tabellenkalkulation) kann je nach gewählten Unterrichtsbeispielen verändert werden. Die Wahl der Schwerpunkte ist auch abhängig von den Vorkenntnissen aus der Grundbildung.

Jgst.	Inhalte	Ziele, Kompetenzen (Methoden)
8.1	<p>Rechnersysteme</p> <ul style="list-style-type: none"> • Aufbau eines Rechners • Hardwarekomponenten • Geschichtliches (Computer u. Informatik) • Betriebssysteme • graphische Benutzeroberflächen • Dateiverwaltungssysteme • Netzwerkstrukturen • LAN • WAN 	<p>Aufbau eines Rechners kennen sachgerechter Umgang mit der Hardware Operationen auf der Ebene des BS Planung und Organisation der Speicherung Anwendungen im Intranet Anwendungen im Internet Einblick und Andeutung der Vielschichtigkeit des Faches Zusammenhänge darstellen und präsentieren können</p>
	<p>Software: Textverarbeitung Formatierung von Texten und Umsetzung verschiedener Möglichkeiten der Gestaltung mit Hilfe einer Textverarbeitungssoftware</p> <ul style="list-style-type: none"> • Formatieren • Tabellen und Objekte einbinden • Aufbau von Briefen • Serienbriefe • Erstellung von Formularen 	<p>Anwenden von Software Reflexion und Bewertung von Textformatierungen Anwenden selbst erstellter „Regeln“ in der Formatierung Computernutzung zur Darstellung und Präsentation (Hausaufgaben, Referate)</p>

	<p>Tabellenkalkulation</p> <ul style="list-style-type: none"> • Rechnen mit Rechenblättern • Arbeiten mit Formeln (rel./absolute Bezüge) • Datentypen • Zuordnungen • grafische Darstellung von Funktionen • Funktionen mit mehreren Parametern • Verkettung von Funktionen • Bedingte Terme • Logische Funktionen • Datenflussdiagramm 	<p>Anwenden von Software Probleme erfassen, erkunden und lösen Mit Daten arbeiten, deren Beziehungen und Veränderungen beschreiben und mit Hilfe des Rechners Ergebnisse bestimmen mathematische Probleme mit dem Computer als Werkzeug lösen Abstrakte Zusammenhänge erkennen, modellieren und mit Hilfe des PCs darstellen</p>
<p>8.2</p>	<p>Einführung in die Software-Entwicklung am Beispiel einer einfachen (ohne Variablen) imperativen Programmiersprache (Niki der Roboter) Basiskonzepte der imperativen Programmierung</p> <ul style="list-style-type: none"> • Prozeduren • Algorithmen mit Verzweigung • Algorithmen mit Wiederholungen • Einführung in die algorithmische Denkweise 	<p>Softwareentwicklung Vom Problem zum Algorithmus Algorithmen und Problemlösestrategien entwickeln und anwenden Modellieren und programmieren</p>
<p>9.1</p>	<p>Erweiterung/Vertiefung der Software-Entwicklung am Beispiel einer imperativen Programmiersprache (Turbo Pascal) Basiskonzepte der imperativen Programmierung</p> <ul style="list-style-type: none"> • Top-Down-Prinzip • Einführung des Variablenkonzeptes • Textmodus, Graphikmodus • Projektarbeit (Textmodus) • Projektarbeit (Graphikmodus) 	<p>Abbilden der Wirklichkeit durch die Darstellung und die Veranschaulichung von Prozessen Die Arbeitsweise eines PCs verstehen und begreifen Modellieren der Wirklichkeit abstrakte Zusammenhänge verstehen und darstellen können Präsentation und Bewertung der Ergebnisse</p>
<p>9.2</p>	<p>Technische Informatik</p> <ul style="list-style-type: none"> • Das EVA-Prinzip • Binäre Zahlencodierung • die Darstellung der Zahlen im Computer • Bit und Byte 	<p>Umsetzung komplexerer Probleme in zielorientierte Steuerungsanweisungen (nach Thema unterschiedlich) Weitere Werkzeuge der Informatik kennenlernen und anwenden können Einblick in vernetzte Zusammenhänge</p>

	<p>Messen, Steuern und Regeln</p> <ul style="list-style-type: none"> • Ansteuerung eines Roboters, Erarbeitung von Algorithmen zur Erledigung bestimmter Aufgaben für einen Roboter • programmieren mit einer graphischen Programmierumgebung (LabView) • Kontroll- und Steuerungsstrukturen 	<p>Praktisches Verständnis der Steuerungsmöglichkeiten von Robotern</p>
	<p>Datenschutz</p>	<p>Weitergabe, Verarbeitung und Nutzung persönlicher Daten; Rechte an eigenen Daten; Schutz persönlicher Daten;</p>

Die obige Reihenfolge der Themen stellt nur eine mögliche Verteilung über die einzelnen Schulhalbjahre dar und die Themenbereichen können in ihrer Reihenfolge zeitlich variiert werden. Je nach Interessenlage der Schüler und vorhandener Zeit kann hier (9.2) ein weiteres Thema gewählt werden, z. B. eine weitere Programmiersprache (Java).

Teil C: Allgemeine Grundlagen S II als Erweiterung zu Teil A (S. 4 – 11)

1. Vorbemerkung

Solange keine aktualisierten Richtlinien und Lehrpläne vorliegen, wird nach der Umstellung auf das G8 mit den bisherigen Lehrplänen weitergearbeitet, wobei der Lehrplan der Jahrgangsstufe 11 (G9) auf die Einführungsphase (G8) übertragen wird; analoges gilt für die höheren Jahrgangsstufen. (s. a. Vorbemerkung zu Teil A)

2. Lernziele des Schullehrplans

Das Schulcurriculum ist in besonderer Weise den im Lehrplan NRW herausgestellten Zielen von Informatikunterricht verpflichtet. Gleichzeitig müssen jedoch – schon im Interesse der Schüler und Schülerinnen – die Anforderungen des Zentralabiturs in die konkrete Planung einbezogen werden.

2.1 Lernziele Informatik NRW

- Fähigkeit, komplexe Zusammenhänge mit gedanklicher Schärfe zu durchdringen (Problemanalyse)
- Schüler kennen typische Verfahren zur Modellbildung in der Informatik
- Trainierung gedanklicher Stringenz, Phantasie und Kreativität (Modellbildung)
- Überblick über unterschiedliche Ansätze beim Problemlösungsprozess
- Sicherer Umgang mit formalsprachlichen Symbolen und Notationen (Implementationsphase)
- theoretische Grundlagen der maschinellen Verarbeitung von Informationen
- Komplexität, Möglichkeiten und Grenzen algorithmischer Verfahren
- Auswirkungen der Informations- und Kommunikationstechnik auf die Lebens- und Arbeitswelt sind bewusst
- Teamfähigkeit, Kommunikations- und Kooperationsfähigkeit
- Urteils- und Kritikfähigkeit bei der Bewertung
- Schüler haben bei der Umsetzung von Problemlösungen und im Rahmen von Projektarbeiten unmittelbare Erfahrungen gesammelt über die Veränderung der individuellen Arbeitsprozesse und der Teamarbeitsprozesse beim Einsatz des Werkzeugs Computer

2.2 Vorgaben zum Zentralabitur

2.2.1 Inhaltliche Schwerpunkte

Grundlegende Modellierungswerkzeuge und Modellierungstechniken

- Grundprinzip des Modellierens als zielgerichtetes Vereinfachen und strukturiertes Darstellen von Ausschnitten der Wirklichkeit
- Imperativer Zugang oder objektorientierter Zugang

Strukturen

- Lineare Strukturen mit den Akzenten
 - Schlange und Stapel mit Standardoperationen - mit Implementation
 - Lineare Liste mit Anwendung der Standardoperationen: Einfügen, suchen, löschen, vor, zurück
 - Sortieren von Listen (mindestens Sortieren durch Einfügen)
 - Quicksort (nur Leistungskurs)
- Baumstrukturen mit den Akzenten
 - Binärbaum: Charakterisierung der Datenstruktur über den rekursiven Aufbau bzw. über die Verknüpfung von Knoten und gerichteten Kanten, Traversierungsalgorithmen
 - Suchbaum als Spezialfall des Binärbaums: Operationen einfügen, suchen und löschen
 - Implementation von Suchen und Einfügen (nur Leistungskurs)
- Ungerichteter gewichteter Graph (nur Leistungskurs) mit den Akzenten
 - Anwendung der Standardoperationen: Knoten bzw. Kante einfügen und löschen
 - Suche des kürzesten Weges zwischen zwei Knoten: Backtracking, Dijkstra-Algorithmus

Endliche Automaten und formale Sprachen

- Darstellung von deterministischen endlichen Automaten als Graph und als Tabelle
- Akzeptor als spezielle Form des endlichen Automaten
- Formale Sprachen – Reguläre Sprachen
- Entwicklung eines Parsers für eine einfache formale Sprache (nur Leistungskurs)

und/oder

Stufen zwischen Hardware und Software

- Funktionsmerkmale eines Prozessors
- Binäre Codierung von Zeichen und Zahlen
- Transformationen auf eine maschinenorientierte Ebene mit den Akzenten
- Zuweisungen: Transport- und arithmetische Befehle
- Verzweigungen und Schleifen: bedingte und unbedingte Sprungbefehle
- Prozeduren - auch rekursiv - mit Parametern auf einer maschinenorientierten Ebene
- Stapel (nur Leistungskurs)

oder

Kommunikation zwischen Computern, Netze

- Schichtenmodell, Protokolle: TCP/IP, HTTP, FTP,..
- Client-Programm: Time-Client oder E-mail-Client
- Client-Server-Programm: Chat-Programm oder Netzwerk-Spiel

Die vorgelegten Materialien werden, soweit es die Syntax betrifft, sowohl in Pascal/Delphi als auch in Java formuliert.

2.2.2 Lern- und Prüfungsbereiche

Unbeschadet der didaktischen und methodischen Freiheit bei der Schwerpunktsetzung und Intensität der Behandlung einzelner Teilgebiete ist darauf zu achten, dass die folgenden Lern- und Prüfungsbereiche für die Abiturprüfung zur Verfügung stehen:

Algorithmusbezogene Qualifikationen

- Algorithmische Grundlagen, Gegenstände und Methoden der Informatik
 - Algorithmen und Datenstrukturen

- Modellierung
- Methoden der Software-Entwicklung

Rechnerbezogene Qualifikationen

- Funktionsprinzipien von Hard- und Softwaresystemen
 - Programmiersprachen und -umgebungen
 - Rechnermodelle und reale Rechnerkonfigurationen
 - Theoretische Grundlagen

Anwendungsbezogene Qualifikationen

- Anwendung von Hard- und Softwaresystemen und deren gesellschaftliche Auswirkungen
 - Anwendungsgebiete
 - Mensch-Maschine-Schnittstelle
 - Grenzen und Möglichkeiten, Chancen und Risiken des Einsatzes der Informations- und Kommunikationstechniken

3. Die besondere Funktion der Einführungsphase

Von der Sekundarstufe I bringen Schülerinnen und Schüler unterschiedliche Vorkenntnisse über wichtige Anwendungsgebiete der Informations- und Kommunikationstechniken mit. Aufbauend auf den bisherigen Erfahrungen und Kenntnissen im Umgang mit dem Computer ist es Ziel der Einführungsphase, die unterschiedlichen inhaltlichen Voraussetzungen der Schülerinnen und Schüler einander anzugleichen, einen Überblick über das Fach Informatik zu liefern und inhaltliche und methodische Grundlagen für das Fach Informatik in der Qualifikationsphase zu schaffen.

Viele Schülerinnen und Schüler haben in der Einführungsphase erstmalig Gelegenheit, sich mit informatischen Fragestellungen auseinander zu setzen, die über den Umgang mit dem Computer und dessen Benutzung hinausgehen. Dem Informatikunterricht kommt deshalb eine wichtige Orientierungsfunktion mit maßgeblicher Bedeutung für die Fortführung des Faches Informatik in der Qualifikationsphase zu.

Manche Schülerinnen und Schüler haben schon im Differenzierungsbereich der Sekundarstufe I Teilaspekte der in der Einführungsphase zu behandelnden Themen kennen gelernt. Im Sinne eines Spiralcurriculums werden dann diese Themen in der Einführungsphase vertieft.

Teil D: Unterrichtspraktischer Teil S II

1. Einführungsphase

1.1. Grundlagen der Programmierung

Für den kompetenten Umgang mit modernen Informatiksystemen benötigt man Hintergrundwissen über Wirkprinzipien und Möglichkeiten automatischer Informationsverarbeitung. Der Zugang zu diesen Inhalten erfolgt über das Thema Grundlagen der Programmierung. Da der Begriff der Programmierung sehr unterschiedlich interpretiert wird, ist eine Klärung nötig. Programmierung wird nicht als ingenieurwissenschaftliche Tätigkeit, sondern ebenso wie das elementare Rechnen in der Mathematik als eine Primärerfahrung mit der Informatik verstanden. Zentrale Begriffe der Informatik erwachsen aus den Erfordernissen des Programmierens; die Erfahrung des Programmierens spielt eine Schlüsselrolle für das Verständnis informatischer Grundbegriffe. Programmierung wird aufgefasst als Problemlösen (Modellieren und Strukturieren) unter Anwendung von Informatikprinzipien und -methoden, als Teilgebiet der Informatik, das die Methoden und Denkweisen beim Entwickeln von Programmen umfasst. Die Programmiersprache ist lediglich Mittel zum Zweck und steht nicht im Zentrum des Unterrichts. Ein Programmierkurs, der von Sprachanstatt von Problemstrukturen ausgeht, verfehlt die Zielsetzungen eines allgemein bildenden Unterrichts.

Programmieren beginnt mit dem Modellieren der betrachteten Problemsituation. Dabei werden für den jeweiligen Zweck die wichtigsten Merkmale und Aktionsmöglichkeiten der beteiligten Objekte unter Vernachlässigung der unwichtigen herausgearbeitet, beschrieben und strukturiert. Daraus ergibt sich das

informatische Modell, das die Rolle eines Bauplans für die Konstruktion eines Informatiksystems spielt. Bei der Modellierung sind äußere Aspekte in Form der Gestaltung der Benutzungsoberfläche zur Ein- und Ausgabe und innere Aspekte wie Darstellung von Informationen durch Daten sowie Verarbeitung von Daten durch Algorithmen zu beachten. Algorithmen können mit Hilfe von Struktogrammen visualisiert werden. Weitere wichtige Hilfsmittel sind Zustands- bzw. Ablaufdiagramme.

Die Konstruktion des informatischen Modells erfolgt mit den Mitteln einer objektorientierten Programmiersprache.

Um an die Erfahrungswelt der Schülerinnen und Schüler anzuschließen, ist der Einsatz eines Entwicklungssystems erforderlich, mit dem Programme für grafikorientierte Benutzungsoberflächen entwickelt werden können. Insbesondere empfehlen sich hierzu Delphi oder Java. Die Fachkonferenz hat sich zur Nutzung der (Java-)Entwicklungsumgebung „NetBeans“ (kostenloser Download unter www.netbeans.org/) entschlossen.

Im ersten Teil der Einführungsphase wird mit den Grundlagen der Programmierung ein Fundament geschaffen, auf dem nun das Thema objektorientierte Modellierung aufbaut. Hier befassen sich die Schülerinnen und Schüler mit Konzepten, Methoden und Verfahren der objektorientierten Modellbildung in den Phasen Analyse, Design und Programmierung. Die objektorientierte Analyse ist Ausgangspunkt der Modellierung. Ein relevanter Ausschnitt der realen Welt wird untersucht und unter Vernachlässigung der unwichtigen auf die bedeutsamen Merkmale reduziert. Der Abstraktionsprozess liefert Attribute, Methoden, Objekte und Klassen, welche in der Unified-Modeling-Language (UML) dargestellt werden. Mittels objektorientiertem Design wird die Struktur eines passenden Informatiksystems entworfen. Strukturelle Beziehungen, zeitliche Abläufe und Datenrepräsentierung sind relevante Gestaltungsaufgaben, deren Lösung sich in Konstruktionsplänen dokumentiert.

Die geplante Struktur des Informatiksystems wird mit der gewählten Programmiersprache in ein lauffähiges Programm übersetzt, mit dem die Lösung getestet werden kann.

An die erfolgreiche Implementierung eines Informatiksystems schließt sich deren Präsentation, die Diskussion der Lösung und eventueller offener Probleme samt ihrer individuellen und gesellschaftlichen Bedeutung sowie die kritische Reflexion des gesamten Lösungsprozesses an. Damit erfahren die Schülerinnen und Schüler ihre Fortschritte in Bezug auf Fach-, Methoden-, Sozial- und Selbstkompetenz.

1.2. Unterrichtsinhalte

1.2.1 Verbindliche Unterrichtsinhalte

Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen

Variablen	Als benannter Behälter für Werte eines bestimmten Datentyps, Wertzuweisung
Einfache Datentypen mit ihren relevanten Operationen und Relationen	Integer, Real, Char, Boolean
Strukturierte Datentypen mit ihren relevanten Operationen und Relationen	String, Feld (array)
Kontrollstrukturen	Anweisungen, Sequenzen. Schleifen, Fallunterscheidungen. Syntaxdiagramme
Struktogramme	grafische Darstellung von Algorithmen
Modularisierung	Prozeduren, Parameter
Benutzeroberfläche	grundlegende Ein-/Ausgabe-Komponenten
Mensch-Maschine-Interaktion	Ereignisse, Ereignisroutinen
Zustandsorientierte Modellierung	Zustände, Übergänge. Zustandsdiagramme
Klassen	Attribute als Datenstruktur zur Repräsentierung der Information über ein Objekt

Methoden als Schnittstellen für den Zugriff auf Attribute und zum Nachrichtenaustausch

Darstellung von Klassen, Objektbeziehungen und Vererbung mit der grafischen Modellierungssprache UML

Vererbung und Klassenhierarchie

Turtlegraphik

1.2.2 Fakultative Unterrichtsinhalte

Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen

Externes Speichern von Zuständen

Textdateien

Funktionale Modellierung

Aufteilung komplexer Systeme in Teilsysteme, Datenflussdiagramme, schrittweise Verfeinerung, Schnittstellen

1.3 Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen

Die Unterrichtsinhalte werden anhand von Themen erarbeitet. Ein themenorientierter Zugang ist an der Verstehensorientierung und auf die Initiierung und Stärkung sachbezogener Motivation ausgerichtet. Er erlaubt exemplarische Vertiefung und umgeht somit das Problem der Stofffülle. Die Themen sind so auszuwählen, dass sie einen Beitrag zur Stiftung kultureller Kohärenz, zur Weltorientierung und zur Entfaltung von Verantwortungsbereitschaft leisten. Sie sollten sich auf authentische Problemsituationen beziehen, die auf Grund ihres Realitätsgehalts und ihrer Relevanz dazu motivieren, neues Wissen oder neue Fertigkeiten zu erwerben. Beispiele für geeignete Themen sind Kalender, Kryptografie, Grafik, Spiele, Ton- oder Bildverarbeitung, Auswertung von Messergebnissen, Simulation technischer Automaten. Auf Grund des interdisziplinären Charakters der Informatik gibt es viele Bezüge zu den anderen Unterrichtsfächern und damit Gelegenheit, auch fachübergreifende und fächerverbindende Themen zu behandeln.

Im Sinne des problemorientierten Lernens sollen Phasen expliziter Instruktion durch die Lehrkraft und konstruktiver Aktivität der Lernenden in einer sinnvollen Balance stehen. Die Schülerinnen und Schüler müssen ausreichend Gelegenheit haben, praktische Erfahrungen im Umgang mit dem Programmentwicklungssystem zu erwerben und Möglichkeiten des selbstständigen Lernens zu nutzen. In der Einführungsphase muss von sehr heterogenen Vorkenntnissen und großen Differenzen in den Lern- und Arbeitstechniken ausgegangen werden. Problemorientiertes Lernen bietet unter diesen Umständen sinnvolle Voraussetzungen für die innere Differenzierung sowie für Partner- und Teamarbeit.

Mit der zustandsorientierten Modellierung werden typischerweise endliche Automaten beschrieben, die genauer in Q1.1 behandelt werden. Da Variablen im Grunde nichts anderes als Programmzustände beschreiben und die Wertzuweisung in imperativen Programmiersprachen im engeren Sinne eine Zustandsänderung darstellt, ist die Verwendung von Variablen immer als zustandsorientierte Modellierung zu verstehen.

Die Schülerinnen und Schüler wenden im Laufe des Kurshalbjahres die Methodik der objektorientierten Modellierung auf ein Problem an, das auf ihr Leistungsvermögen und den zur Verfügung stehenden Zeitrahmen abgestimmt ist.

Am Ende der Einführungsphase kann ein kleines Projekt dazu dienen, in Teamarbeit und arbeitsteiliger Vorgehensweise bei der Analyse von Problemen, der Erstellung und dem Test von Lösungen sowie der Dokumentation des Lösungsprozesses einzuführen.

2. Q1

2.1 Q1/1: Konzepte und Anwendungen der Theoretischen Informatik

2.1.1 Begründung

Die Schülerinnen und Schüler haben in der Einführungsphase in Ansätzen Elemente der theoretischen Informatik kennen gelernt. Die vertiefte Auseinandersetzung mit relevanten theoretischen Konzepten in diesem Kurshalbjahr baut auf diesen Vorkenntnissen auf und stellt Bezüge zu den im Informatikunterricht behandelten Themen her. Die Theorie ordnet Erfahrungen durch Systematisierung, weist Zusammenhänge auf und begründet bzw. beweist sie. Sie schafft durch ihre Begrifflichkeit die Voraussetzungen, um über die prinzipiellen Grenzen von Technik und menschlichem Geist etwas aussagen zu können. Die Ergebnisse der Theorie sind von beständigem Wert und nicht den schnellen Entwicklungszyklen und kurzen Halbwertszeiten der hektischen Computerwelt unterworfen.

Die Inhalte der theoretischen Informatik sind nicht immer so direkt anwendbar wie viele Inhalte anderer informatischer Themenbereiche. Dafür sind die Erkenntnisse der theoretischen Informatik oft allgemeiner, umfassender und weitreichender als in anderen Gebieten der Informatik. Gemäß dem Grundsatz „Nichts ist praktischer als eine gute Theorie“ gilt es, die Bedeutung der Theorie durch deren Anwendung in der Praxis auf Schulniveau zu verdeutlichen. Die Behandlung theoretischer Fragestellungen darf nicht isoliert von den Anwendungen und auf Vorrat erfolgen. Übertriebene Formalisierungen gilt es zu vermeiden.

Die Inhalte sind didaktisch reduziert darzubieten, ohne dabei die wesentlichen Einsichten zu verfälschen. Die Informatik setzt sich mit informationsverarbeitenden Maschinen auseinander. Computersysteme unterstützen und erleichtern die geistige Arbeit von Menschen. Daher ist ein fundiertes Grundwissen über die Möglichkeiten und Grenzen der Organisation von Arbeit durch den IT-Einsatz zu vermitteln. Die Technische Informatik geht der Frage nach, wie informationsverarbeitende Maschinen gebaut werden können und wie die Hardware solcher Maschinen aussieht. Entsprechende Unterrichtsinhalte können im fakultativen Bereich behandelt werden. Die Theoretische Informatik setzt sich mit der Leistungsfähigkeit, den prinzipiellen Möglichkeiten und den Grenzen informationsverarbeitender Maschinen sowie den Grenzen der Problemlösung mit Algorithmen auseinander.

Die Theorie der Automaten und formalen Sprachen spielt eine wichtige Rolle bei der Entwicklung von Sprachen, z. B. Auszeichnungs-, Programmier- oder Datenbanksprachen, sowie von entsprechenden Übersetzer- und Interpreterprogrammen. Schülerinnen und Schüler haben in ihrer bisherigen Auseinandersetzung mit der Informatik mit diesen Programmen schon gearbeitet; nun findet eine Untersuchung ihrer Wirkprinzipien statt. Formale Sprachen sind Mittel der Kommunikation zwischen Mensch und Maschine. Der Zusammenhang zwischen formaler Sprache und erkennendem Maschinenmodell führt zur Chomskyschen Sprachhierarchie und damit zur Charakterisierung der formalen Sprachen durch Typen von Grammatiken.

Endliche Automaten als abstrakte Modelle realer Computer bilden einen möglichen Zugang zu theoretischen Grundlagen der Informatik. An ihnen lässt sich sowohl der Zusammenhang mit den von ihnen akzeptierten Sprachen als auch die prinzipielle Begrenztheit ihrer Leistungsfähigkeit zeigen. Kellerautomaten und Turingmaschinen stellen Automaten höherer Leistungsfähigkeit dar. Die Turingmaschine als universelle symbolverarbeitende Maschine repräsentiert nicht nur eine Präzisierung des Algorithmenbegriffs, sondern dient auch zur Veranschaulichung der prinzipiellen Grenzen maschineller Informationsverarbeitung.

Die Berechenbarkeitstheorie beschäftigt sich mit den grundsätzlichen Möglichkeiten und Grenzen der Algorithmisierbarkeit. Historisch gesehen steht sie am Anfang der Informatik, noch vor der Computer-Ära. Haupteinsicht dieser Theorie ist die Churchsche These, wonach die Klasse der im intuitiven Sinne berechenbaren Funktionen mit der Klasse der turing-berechenbaren Funktionen übereinstimmt und es nichtberechenbare Funktionen bzw. algorithmisch unlösbare Probleme gibt.

Im Grundkurs sollen die Fachbegriffe und Zusammenhänge möglichst anschaulich eingeführt werden, vertiefte mathematische Formalismen und Methoden sind zu vermeiden. Soweit möglich sind die theoretischen Konzepte mit konkreten Anwendungen zu verbinden.

Die Modellierung einfacher realer Automaten (z. B. Getränkeautomat, Fahrkartenautomat, Aufzugmodell) führt zu den endlichen Automaten, die sich grafisch als Zustandsdiagramme darstellen lassen. Pfade auf dem

Zustandsdiagramm werden durch die vom Automaten akzeptierten Wörter beschrieben. Die reguläre Grammatik ergibt sich aus der Umsetzung einzelner Zustandsübergänge in Produktionen. Zu den Automaten können die Schülerinnen und Schüler eigene Simulationsprogramme schreiben oder sie mit verfügbaren Programmen auf der Basis der Grammatiken simulieren.

Einfache Beispiele wie z. B. Scanner für Bezeichner, Zahlen oder Kontrollstrukturen führen schnell zu kontextfreien Grammatiken für reguläre Sprachen. Syntaxdiagramme schließen an die Vorkenntnisse der Schülerinnen und Schüler an und können als anschauliche Darstellungsmittel für Grammatiken genutzt werden.

Anwendungsbeispiele stellen den Lebensweltbezug her. So können beispielsweise Elemente der bisher behandelten Sprachen in Syntaxdiagrammen oder als Grammatiken dargestellt werden.

Die in RFCs (Request for Comment) festgelegten Internetstandards bieten weitere Anwendungsbeispiele (z. B. RFC 821 – Simple Mail Transfer Protocol, RFC 1945 – Hypertext Transfer Protocol). Der Bereich der regulären Sprachen wird in den Beispielen im Allgemeinen erst dann verlassen, wenn Klammerterme bzw. rekursive Syntaxdiagramme auftreten.

Beim Thema Berechenbarkeit geht es um den Nachweis, dass es nicht berechenbare Probleme gibt, und um die Unentscheidbarkeit des Halteproblems. Weitere Probleme, die auf das Halteproblem rückführbar sind, können behandelt werden.

Der Vergleich von Algorithmen mit polynomialer bzw. exponentieller Laufzeit führt schnell auf die Begriffe berechenbar und durchführbar. Erheblich mehr Aufwand ist für den Nachweis der Existenz nicht berechenbarer Funktionen nötig; schließlich muss man das Cantorsche Diagonalisierungsverfahren bemühen. Dagegen ist der Nachweis der Unentscheidbarkeit des Halteproblems mit deutlich weniger mathematischem Aufwand möglich.

Die Turingmaschine hat eine einfachere Struktur als eine Registermaschine. Letztere sind aber leichter mit realen Mikroprozessoren vergleichbar. Exemplarisch kann man zeigen, dass mit Turingmaschinen arithmetische und logische Ausdrücke sowie alle Kontrollstrukturen verarbeitet werden können und sie somit genauso berechnungsstark wie Registermaschinen und moderne Computer sind.

Für die aktive Auseinandersetzung mit den theoretischen Maschinenmodellen sind Simulationen nützlich. Simulationsprogramme sind auch im Internet verfügbar.

2.1.2 Unterrichtsinhalte

2.1.2.1 Verbindliche Unterrichtsinhalte **Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen**

Formale Sprachen und Grammatiken	reguläre und kontextfreie Grammatiken und Sprachen Anwendung mit Syntaxdiagrammen Chomsky-Hierarchie (LK) kontextsensitive Sprachen (LK)
Endliche Automaten	Zustand, Zustandsübergang, Zustandsdiagramm, Zeichen, Akzeptor Simulation realer Automaten (z. B. Getränkeautomat) Anwendung endlicher Automaten (z. B. Scanner) deterministische und nicht-deterministische Automaten (LK) reguläre Ausdrücke (LK) Mensch-Maschine-Kommunikation (LK)
Kellerautomaten (LK, GK fakultativ)	Automat mit Kellerspeicher kontextfreie Grammatiken Klammerausdrücke, Rekursion
Turing- oder Registermaschine (LK, GK fakultativ)	Turing- oder registerberechenbar Churchsche These Computer als universelle symbolverarbeitende Maschine

	Verhältnis Mensch – Maschine
Berechenbarkeit	Entscheidbarkeit, Halteproblem prinzipielle Grenzen algorithmischer Verfahren
2.1.2.2 Fakultative Unterrichtsinhalte	Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen
Übersetzerbau	Scanner, Parser, Interpreter und Compiler z. B. Steuersprache für Roboter, LOGO, Plotter oder miniPASCAL
Technische Informatik	Rechnerarchitektur, Assemblersprache (vgl. Wahlthema in Q2)
Komplexitätstheorie	die Klassen P und NP, P/NP-Problematik NP-vollständige Probleme

2.2 Q1/2: Objektorientierte Modellierung und Datenstrukturen

In Bezug auf Algorithmen findet eine Vertiefung statt. Die effiziente Informationsverarbeitung erfolgt auf der Basis von Standardalgorithmen. Die Rekursion als wichtiges Programmierkonzept wird anhand von Beispielen mit der Iteration verglichen. Beispiele aus dem Bereich der Grafik ermöglichen eine unmittelbare Rückmeldung über die Korrektheit der Lösung. Im Grundkurs genügt die Behandlung elementarer Algorithmen zum Suchen und Sortieren. Im Leistungskurs sind auch effiziente Algorithmen nach dem Prinzip Teile und Herrsche bzw. Backtracking zu behandeln.

Die Verarbeitung großer Datenmengen und flexibler Datenstrukturen erfordert die Behandlung abstrakter Datentypen. Im Sinne objektorientierter Modellierung entwirft und implementiert man abstrakte Datentypen als Klassen. In Attributen werden die Daten gespeichert; Standardalgorithmen erscheinen als Methoden der jeweiligen Klasse.

Moderne Programmentwicklungssysteme bieten fertige Klassen für die zu betrachtenden abstrakten Datentypen an, so dass diese nicht selbst entwickelt werden müssen. Unter diesem Aspekt spielen eher die Orientierung in der zum Entwicklungssystem gehörenden Klassenhierarchie, die Fähigkeit, die für die eigene Problemlösung relevanten Klassen und Methoden zu finden, und die Anwendung durch Ableitung spezialisierter Klassen, z. B. für spezielle Datentypen, eine wichtige Rolle.

Demgegenüber spielt unter dem Aspekt der eigenen Konstruktion das Zeigerkonzept zur Realisierung dynamischer Datenstrukturen eine wichtige Rolle. Die explizite Arbeit mit Zeigern ist nicht unbedingt erforderlich.

Ein Einblick in das Zeigerkonzept als Variablen mit dem Variablentyp Adressen genügt. Die abstrakten Datentypen Keller und Schlange können beispielsweise als spezielle lineare Listen aufgefasst und daher von einer vorgegebenen Listen-Klasse abgeleitet werden.

2.2.1 Unterrichtsinhalte

2.2.1.1 Verbindliche Schwerpunkte:	Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen
Standardalgorithmen	rekursive und iterative Verfahren einfache Such- und Sortierverfahren binäre Suche (LK)
Abstrakte Datentypen (LK, GK fakultativ)	Repräsentierung und Standardoperationen Keller, Schlange lineare Liste, binärer Suchbaum
Effiziente Algorithmen (LK, GK fakultativ)	schnelle Sortierverfahren Suchen durch Backtracking Teile- und Herrsche-Prinzip
Komplexität von Algorithmen	polynomiale und exponentielle Zeitkomplexität

(LK, GK fakultativ)

Grundkonzepte des Software-Engineerings Problemanalyse. Modellierung. Entwurf, Implementation, Test, Revision, Dokumentation

2.2.2 Arbeitsmethoden der Schülerinnen und Schüler/Hinweise und Erläuterungen

Je nach Erfahrung mit Projektarbeit arbeiten sie in Teams an verschiedenen Projekten oder gemeinsam an einem Projekt. Die Bearbeitung fachübergreifender Probleme bietet sich an, wobei möglichst die behandelten abstrakten Datentypen und Algorithmen zum Einsatz kommen sollen. Die Dokumentation des Projekts erfolgt sinnvollerweise in Form von HTML-Dokumenten (siehe Grundkurs EF). Bei Verwendung von Java kann das Projektergebnis als Applet direkt in die Dokumentation eingebunden werden.

Zur Vernetzung des Wissens sind die Bezüge zum Kurshalbjahr Q1 wichtig. Der abstrakte Datentyp Keller und die Methode der Rekursion können zur Simulation von Kellerautomaten benutzt werden. Erfahrungsgemäß gestaltet sich die Konstruktion eines Kellerautomaten zu einer kontextfreien Sprache schwierig, weswegen in der Literatur oft nur einfache Beispiele zu kontextfreien Sprachen vorkommen. Ein geeignetes Mittel ist beim Einsatz von Syntaxdiagrammen die Ersetzung rekursiver Aufrufe durch push-/pop- Operationen.

Die Vorgabe einer Struktur für die Dokumentation ist hilfreich. Die erforderlichen UML-Diagramme können mit Grafikwerkzeugen hergestellt werden. Die Qualität der Dokumentation ergibt sich unter anderem durch die Klarheit im Aufbau, die Konzentration auf das Wesentliche, die verständliche und fachlich einwandfreie Beschreibung, den expliziten Bezug zwischen Text, Bild und Diagrammen, die kritische Diskussion der Ergebnisse und die Reflexion der Auswirkungen beim Einsatz des Informatiksystems.

Zum Abschluss der Projektphase stellen die Teams ihre Arbeit in Form einer Präsentation im Plenum vor. Die Präsentation hat in Bezug auf die Dokumentation ihren eigenen Stellenwert, der sich aus einem Vortrag in Kombination mit Präsentationsfolien oder HTML-Seiten ergibt. Für die Beurteilung der Präsentation sind auch kommunikative Fähigkeiten, angewandte Präsentationstechnik und bewusste Körpersprache heranzuziehen.

3. Q2

Für die Kurse im zweiten Teil der Qualifikationsphase sind Wahlthemen vorgesehen. Dabei sind insbesondere die für das Zentralabitur des jeweils aktuellen Jahres festgelegten thematischen Schwerpunktsetzungen zu beachten. Angesichts der besonderen Situation im Prüfungshalbjahr ist es nicht notwendig, mit dem Wahlthema ein völlig neues Sachgebiet zu beginnen. Sinnvoll sind Wahlthemen, die an die in den vorhergehenden Halbjahren gelegte Basis anschließen. Die zur Verfügung stehenden Wahlthemen reichen vom eher hardware-orientierten Ansatz über die Untersuchung spezieller Grundlagen bis hin zu aktuellen Anwendungsgebieten. In jedem Wahlthema ergeben sich Verknüpfungen zu anderen Wissensgebieten, so dass hier geeigneter Raum zur Diskussion und Untersuchung philosophisch-ethischer und fächerverbindender Aspekte oder dem Aufzeigen gesellschaftspolitischer oder wissenschaftlich relevanter Zusammenhänge ist.

Zur Vorbereitung auf das Abitur sollen bisher erworbene Kenntnisse in neuen Zusammenhängen verwendet oder vertieft und Querverbindungen zwischen den behandelten Themen erarbeitet und genutzt werden. Auf eine Wiederholung und Anwendung bekannter Verfahren darf sich der Kurs nicht beschränken.

3.1. Beispiele

3.1.1 Q2-Wahlthema: Computergrafik

3.1.1.1 Begründung

Am Anfang des Kurses wird sicher eine Phase notwendig sein, in welcher die zur Verfügung stehenden Möglichkeiten im grafischen Bereich untersucht werden. Dieses Thema bietet dann reichhaltige Möglichkeiten, sich mit verschiedenen Teilgebieten der Informatik zu befassen. Grafische Objekte lassen sich als Primitive definieren, die objektorientiert realisiert werden. Beim Übergang zu dreidimensionalen Grafiken kommen Anwendungen der Matrizenrechnung zum Zuge. Der Begriff der Perspektive lässt unterschiedliche Methoden der Umsetzung zu. Anspruchsvolle Clipping-, Hidden- Surface- oder Komprimierungs-Algorithmen können behandelt werden.

Die Grundlagen und Anwendungen in diesem Gebiet der Informatik geben weiterhin die Gelegenheit für fächerverbindenden oder fachübergreifenden Unterricht. Mathematik oder Kunst bieten sich an.

3.1.1.2 Unterrichtsinhalte

- Computergrafik in 2 Dimensionen Zeichenwerkzeuge und -möglichkeiten
- Grafische Primitive (Kreis, Quadrat, ...)
- Repräsentation von Objekten
- Bewegung
- Clipping
- 3D-Modelle Transformation 3D auf 2D
- Perspektive
- Optimierungsstrategien Hidden Surface
- Farbmodell RGB, CMYK, 24-Bit-Farbraum, Paletten
- Komprimierung JPEG, GIF (LZW-Komprimierung) oder TIFF (LZH-Komprimierung)
- Lichtquellen Raytracing
- Radiosity
- Softwareschnittstellen Fluchtpunkte
- Softwareschnittstellen DirectX
- OpenGL

3.1.2 Q2-Wahlthema: Simulationen

3.1.2.1 Begründung

Ist eine Möglichkeit zum fächerverbindenden Arbeiten gegeben, kann das Thema Simulationen ins Auge gefasst werden. Simulationen nehmen einen breiten Raum in den Anwendungen der Informatik ein. Ihre Grundlagen bestehen im Allgemeinen sowohl aus einem mathematischen wie auch einem fachwissenschaftlichen Anteil. Das Räuber-Beute-Modell in der Biologie auf der einen Seite, die Entwicklung finanzmathematischer Zusammenhänge und die Betrachtung von Fraktalen am anderen Ende zeigen das Spektrum der Möglichkeiten, welche die Behandlung dieses Themas bietet.

Die informatischen Inhalte in diesem Gebiet können sich unter anderem aus der Algorithmik, der Optimierung des Laufzeitverhaltens und der grafischen Ausgabe rekrutieren. Auch die Umsetzung spezieller mathematischer Methoden ist ein lohnendes Gebiet, welches wieder in das Feld der Datenstrukturen und Objekte führen kann. Fächerverbindend mit beispielsweise Mathematik, Biologie oder Wirtschaftswissenschaften ist folgende exemplarische Auswahl von Inhalten realisierbar:

3.1.2.2 Unterrichtsinhalte

Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen

Modellbildung	Realität → Modell → Simulation
Abgeschlossenes System	Räuber-Beute-Modell Differenzgleichungen Grafische Umsetzung
Rückkopplungskreise	Ursache-Wirkungs-Diagramme Eskalierender und stabilisierender Kreis Bevölkerungsmodell als Kombination Begrenzttes Wachstum
Chaotische Systeme	Komplexe Zahlenfolgen/Iterationsverfahren Mandelbrot-Menge Fraktale Grafische Darstellung

3.1.3 Q2-Wahlthema: Rechnerstrukturen

3.1.3.1 Begründung

Grundlagen aus dem Themenbereich der Technischen Informatik leisten einen Beitrag für das Verständnis der Funktionsprinzipien von Informatiksystemen als symbolverarbeitende, universelle Maschinen. Im Vordergrund steht die exemplarische Auseinandersetzung mit den Ideen und den Prinzipien der Computersysteme, die sie als technisches Produkt ermöglichen. Dabei soll ausgehend von den historischen Entwicklungslinien und ihrer Realisierung auf unterschiedlichen technischen Ebenen abstrahiert werden. Chancen, Risiken und Folgen bei der Entwicklung zur Informationsgesellschaft und der sich damit verändernden Lebens- und Arbeitsformen lassen sich hierbei aufzeigen.

In der Regel wird man mit den logischen Grundsaltungen beginnen und mit Volladdieren in Form von Blockschaltbildern ein einfaches Addier- und Subtrahierwerk mit Komplementbildung realisieren. Die Schaltnetze können gut mit Simulations-Programmen aufgebaut und analysiert werden. Nach der Behandlung von einfachen Speicher-Bausteinen wird man das „Von Neumann Prinzip“ als Abstraktion und Grundlage moderner Mikroprozessoren einführen. Anhand des Blockschaltbildes und von Simulationen eines Mikroprozessors (z. B. 8086) werden dessen Grundprinzipien erarbeitet. Der Kurs soll mit der Analyse einiger beispielhafter Assemblerprogramme abgerundet werden.

3.1.3.2 Unterrichtsinhalte

Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen

Logische Grundsaltungen	AND, OR, NOT, NAND, NOR, EXOR, Arbeitstabelle, Symbol
Binärcodierung	Binärcode, BCD-Code, HexCode, Subtraktion durch Komplementbildung
Einfache Rechenwerke	Halbaddierer, Volladdierer, 4-Bit Addier- und Subtrahierer
Speicherelemente	RS-Flip-Flop, D-Flip-Flop, JK-Master-Slave-Flip-Flop, Anwendungen, DEA
Von-Neumann-Prinzip	Funktionseinheiten, Speichermodell, Befehlsabarbeitung
Mikroprozessor- und Mikrocomputerarchitektur	Bsp.: SL/16, Bussysteme, ALU, Register, Flag-Register, Steuerwerk, Mikroprogramm, BIOS, ROM, RAM
Assemblerprogrammierung	Entwicklungswerkzeuge, Beispiele, Registermaschinen, Adressierungsarten, Codeschablonen für Schleifen und Fallunterscheidungen, Interpreter
Technische Realisierung von Grundsaltungen	Relais-, Transistorschaltungen

3.1.4 Q2-Wahlthema: Rechnernetze

3.1.4.1 Begründung

Lokale und globale Rechnernetze bilden die Grundlage der Informations- und Kommunikationssysteme am Übergang vom zweiten zum dritten Jahrtausend. Das ISO/OSI-Referenzmodell liefert für die Auseinandersetzung mit dem Thema Rechnernetze einen sinnvollen Rahmen. Mit ihm kann das Prinzip der Rechnerkommunikation gut beschrieben werden. Im Unterrichtsverlauf wird man Schichten des Referenzmodells untersuchen und einzelne Aspekte vertiefend behandeln.

TCP/IP als Grundlage des Internets ist den Schülerinnen und Schülern schon aus E1 bekannt, so dass hier genauer zwischen den einzelnen Ebenen unterschieden werden soll. Sofern noch nicht geschehen, können die Schülerinnen und Schüler eigene Internetanwendungen auf der Basis von Sockets erstellen.

Technische Aspekte spielen bei der Betrachtung von Netzwerkstrukturen eine Rolle. Da verschiedene Medien (wie Kupfer, Glasfaser und Funk) in jedem Netz kombiniert werden können, müssen Übergangsmöglichkeiten

geschaffen werden. Mit Hilfe des ISO/OSI-Referenzmodells lassen sich sowohl die verschiedenen Probleme als auch deren Lösungen gut darstellen.

Angesichts der Gefährdungen durch das Internet sollten auch sicherheitsrelevante Aspekte betrachtet werden. An einer Firewall lassen sich Schutzmöglichkeiten für lokale Netze aufzeigen. Wurde ein SQL-Server beim Thema Datenbanken nicht behandelt, bietet sich hier eine gute Gelegenheit dazu.

3.1.4.2 Unterrichtsinhalte

Kommunikationssysteme
Prinzipien und Protokolle
Dienste und Anwendungen
Netzwerke

Routing
Sicherheit
Weitere Dienste und Anwendungen
Server

Bezüge zur Obligatorik, methodische Schwerpunkte, Kompetenzen

ISO/OSI-Referenzmodell
CSMA/CD TCP und IP, UDP
Kommunikation über Sockets und Ports, POP3, SMTP, HTTP, FTP
Topologien: Stern, Bus, Ring
Arten und Medien: Ethernet (802.3, 802.11),
BNC-Kabel, RJ45 (Twisted-Pair)-Kabel, Lichtwellenleiter, Funk
Komponenten: Hubs und Switches
Router, NAT
Prinzip des Firewalls
News, Timeserver, Proxies
SQL-Server