

Rahmenrichtlinien
Gymnasium
Informatik

An der Überarbeitung der vorläufigen Rahmenrichtlinien haben mitgewirkt:

Eschrich, Mario	Halle
Dr. Gutzer, Hannes	Halle (betreuender Dezernent des LISA)
Dr. Herper, Henry	Magdeburg (fachwissenschaftlicher Berater)
Dr. Lehmann, Hans	Osterburg
Zuber, Jörn	Blankenburg

Vorwort

Inhaltsverzeichnis

	Seite
1 <u>Aufgaben des Grundkurses Informatik am Gymnasium</u>	6
2 <u>Qualifikationen und fachdidaktische Konzeption</u>	8
3 <u>Zur Arbeit mit den Rahmenrichtlinien</u>	11
4 <u>Grundsätze der Unterrichtsgestaltung</u>	12
4.1 <u>Didaktische Grundsätze</u>	12
4.2 <u>Leistungen und ihre Bewertung</u>	13
5 <u>Inhalte</u>	14
5.1 <u>Übersichten</u>	14
5.1.1 <u>Themen und Kurse in der Sekundarstufe II</u>	14
5.2 <u>Darstellung der Themen in der Sekundarstufe II</u>	16
5.2.1 <u>Schuljahrgang 11 (Einführungsphase)</u>	16
5.2.2 <u>Schuljahrgänge 12/13 (Qualifikationsphase)</u>	28

1 Aufgaben des Grundkurses Informatik am Gymnasium

Der Informatik als Wissenschaft kommt beim Übergang zu einer Informationsgesellschaft eine Schlüsselrolle zu, da sie systematisch Möglichkeiten der automatischen Informationsverarbeitung und Wissensrepräsentation erforscht und in Informatiksystemen nutzbar macht.

Der Grundkurs Informatik am Gymnasium ist hinsichtlich seines Gegenstandes und seiner Lern- und Arbeitsmethoden als integrativer Bestandteil des mathematisch-naturwissenschaftlichen Unterrichts anzusehen. Seine Aufgaben gehen im Rahmen des fächerübergreifend zu betrachtenden Bildungs- und Erziehungsprozesses über den nahe liegenden Bezug zu den naturwissenschaftlichen Fächern hinaus. In der Sekundarstufe I werden im Rahmen der Informations- und Kommunikationstechnologischen Grundbildung (IKG) und in Wahlpflichtkursen (zum Beispiel Moderne Medienwelten) Grundlagen zur Nutzung des Computers in Verbindung mit Standardsoftware vermittelt. Die dynamische Entwicklung der Informatik und Informationstechnologie und deren Anwendung bedingen eine ständige Erweiterung und Wichtung der Unterrichtsinhalte und Arbeitsmethoden.

Die Aufgabe des Informatikunterrichtes in der Sekundarstufe II besteht darin, Schülerinnen und Schülern einen Zugang zu informatischen Denk- und Arbeitsweisen sowie Informatiksystemen zu öffnen. Informatische Bildung ist das Ergebnis von Lernprozessen, in denen Grundlagen, Methoden, Anwendungen, Denk- und Arbeitsweisen sowie die gesellschaftliche Bedeutung von Informatiksystemen Unterrichtsgegenstände sind und damit in Hinblick auf das wissenschaftspropädeutische Arbeiten in der Kursstufe bedeutsam. In allen Phasen der Lernprozesse im Grundkurs Informatik wird die Vermittlung von grundlegendem Wissen der informatischen Bildung realisiert. Dabei werden Grundkenntnisse aus den Bereichen der theoretischen Informatik, technischen Informatik, angewandten Informatik sowie deren gesellschaftliche Auswirkungen vermittelt. Damit wird auch deutlich, dass jedes Informatiksystem als Kombination von Hard- und Software-Komponenten das Ergebnis des informatischen Modellierens ist. In allen Phasen der informatischen Bildung ist die Informatik die Bezugswissenschaft. Vielfältige und wechselseitige Beziehungen bestehen zu Gegenständen der anderen Naturwissenschaften, der Technik, aber auch zu den gesellschaftswissenschaftlichen Fächern. In diesem Rahmen

ordnet sich das Fach Informatik in das Gesamtkonzept zum fächerübergreifenden Unterricht ein.

Hierbei werden die Vorteile aber auch die Risiken, Gefahren und Grenzen mit den gesellschaftlichen Auswirkungen bei der Nutzung von Informatiksystemen deutlich. Das Erkennen und das Analysieren von Informatiksystemen in der realen Welt gehört damit zu den grundlegenden Aufgaben des Grundkurses Informatik.

2 Qualifikationen und fachdidaktische Konzeption

Qualifikationen

Der Informatikunterricht als Kern einer allgemeinbildenden informatischen Bildung in den Jahrgangsstufen 11-13 zielt auf die Herausbildung solcher Qualifikationen ab, die gegenwärtig und zukünftig bei der Arbeit mit Informatiksystemen unverzichtbar sind.

Die Schülerinnen und Schüler

- können Informationen beschaffen, strukturieren, bearbeiten, speichern, darstellen, bewerten und in geeigneter Form präsentieren (Methodenkompetenz),
- sind in der Lage, miteinander zu lernen, zu kommunizieren und selbst als Mitglied einer Lerngruppe Verantwortung zu übernehmen sowie Kritik zu ertragen und konsensfähige Lösungen anzustreben (Sozialkompetenz),
- erfahren im Umgang mit Informatiksystemen persönlich bedeutsame Werte und entdecken dabei individuelle Neigungen, Begabungen und Interessen (Selbstkompetenz),
- besitzen Kenntnisse zu ausgewählten Methoden und Verfahren der Informatik und erhalten einen Einblick in formale Konzepte der Informatik,
- verfügen über Fähigkeiten und Fertigkeiten bei der Lösung von Aufgaben durch die Anwendung von Informatiksystemen,
- beherrschen grundlegende Denk- und Arbeitsweisen der Informatik, die sie sich durch Anwendung, Analyse, Modifikation und Bewertung ausgewählter Informatiksysteme angeeignet haben,
- können fächerübergreifend die grundlegenden Denk- und Arbeitsweisen der Informatik mit den Inhalten anderer Fächer verknüpfen,
- besitzen Kenntnisse und Fähigkeiten zur Auseinandersetzung mit gesellschaftlichen, sozialen und ethischen Grundfragen,
- können mit produktbezogenen Hilfesystemen arbeiten und sind in der Lage, sich eigenständig in unbekannte Systeme einzuarbeiten,
- haben grundlegende Kenntnisse über Aufbau und Einsatz von Informatiksystemen,
- können Informatiksysteme bezüglich ihrer Funktion und Struktur historisch einordnen,
- kennen theoretische Grundlagen der Arbeitsweise von Computern,

- kennen wesentliche Funktionen von Betriebssystemen und können diese anwenden,
- kennen grundlegende Funktionen und Strukturen von Rechnernetzen und deren theoretische Grundlagen,
- können Algorithmen entwerfen und diese mit einer Programmiersprache implementieren,
- besitzen umfassende Kenntnisse in einem der angebotenen Wahlthemen,
- kennen die Phasen des informatischen Modellierens und können diese an einem Projekt umsetzen.

Fachdidaktische Konzeption

Die in den Aufgaben des Faches Informatik am Gymnasium charakterisierte informatische Bildung orientiert sich an den nachstehenden Leitlinien

- Interaktion mit Informatiksystemen,
- Wirkprinzipien von Informatiksystemen,
- Informatische Modellierung und
- Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft.

Die unter diesen Leitlinien strukturierten Kenntnisse und Fertigkeiten werden, auf unterschiedlichem Ausgangsniveau basierend, im Informatikunterricht in den Jahrgangsstufen 11 bis 13 erworben, wobei stets an die Erfahrungs- und Lebenswelt der Schülerinnen und Schüler anzuknüpfen ist. Der Informatikunterricht ist so zu gestalten, dass die Schülerinnen und Schüler durch vielfältige geistige Tätigkeiten und praktische Arbeiten mit Informatiksystemen ein anwendungsbereites Wissen im Umgang mit digital dargestellten Informationen und die Beherrschung von Informatiksystemen als unverzichtbare Ergänzungen der traditionellen Kulturtechniken erwerben. Zu diesen zählen

- die Beschaffung von Informationen,
- die Darstellung von Informationen in maschinell verarbeitbaren Zeichen (Daten),
- die maschinelle Verarbeitung und Verteilung der Daten und
- die Gewinnung neuer Informationen durch Interpretation der gewonnenen Daten, die zusammen mit dem Vorwissen zu neuem Wissen führt.

Der Gegenstand der Informatik als junge und dynamische Wissenschaft stellt in Anbetracht des zur Verfügung stehenden Zeitvolumens hohe Anforderungen an die didaktische Reduktion der Lerngegenstände und der weitgehend exemplarischen Behandlung. Die Rahmenrichtlinien im Schuljahrgang 11 enthalten eine Einführungsphase, um die Schaffung eines gesicherten Basisniveaus für alle Schülerinnen und Schüler zu ermöglichen. Hierzu ist eine starke Differenzierung im Unterricht notwendig. Die vorliegenden Rahmenrichtlinien gehen davon aus, dass jeder Schülerin und jedem Schüler ein eigenständiger Computerarbeitsplatz zur Verfügung steht.

Die im 11. Schuljahrgang behandelten zentralen Themen „Datenschutz und Datensicherheit“ und „Computernetzwerke“ haben eine besondere Bedeutung für den Informatikunterricht, da Computer heute in der Regel keine eigenständigen Arbeitsstationen, sondern Bestandteil von lokalen und globalen Netzwerken sind und in vielen gesellschaftlichen Bereichen Einzug gehalten haben.

In der Qualifikationsphase erlernen die Schülerinnen und Schüler die grundlegenden Techniken des informatischen Modellierens als eine wesentliche Grundlage für die Entwicklung von Informatiksystemen. Das schließt das Erlernen einer Programmiersprache ein. Da die Informatik über ein sehr breites Spektrum an Wissensgebieten verfügt, werden im Kurshalbjahr 13/1 Wahlthemen angeboten. Die Auswahl des Themas sollte sich an den Interessen der Schülerinnen und Schüler sowie an den schulischen Gegebenheiten orientieren. Entsprechend der Profilierung der Schule oder auf Grund aktueller Entwicklungen der Fachwissenschaft Informatik ist es möglich, abweichend von den vorgegebenen Wahlthemen ein eigenes Thema zu entwickeln, das sich an den vorgegebenen Zielen und an dem vorgegebenen Niveau der anderen Wahlthemen orientieren muss. Die Projektarbeit im Kurshalbjahr 13/2 dient der Vertiefung der angestrebten Qualifikationen und Kompetenzen.

3 Zur Arbeit mit den Rahmenrichtlinien

Die Rahmenrichtlinien stellen die Grundlage für die Gestaltung des gesamten Informatikunterrichtes im Sekundarbereich II dar. Für die schulische Planungsarbeit bilden die Ziele und Qualifikationen den verbindlichen Rahmen.

Die in den Schuljahren 11 und 12 vorgegebenen Themen sind Pflichtthemen. Die Verbindlichkeit bezieht sich auf die Themen, die zugeordneten Ziele und Qualifikationen sowie die jeweiligen Inhalte. Die Reihenfolge der Themen innerhalb der Jahrgänge 11 und 12 kann von der Lehrkraft verändert werden, soweit es die fachliche Logik zulässt. Die themenbezogenen Zeitrichtwerte sind Empfehlungen. Die verbleibende Unterrichtszeit sollte

- zur ergänzenden bzw. vertiefenden Behandlung von Inhalten entsprechend der Interessenlage der Schülerinnen und Schülern,
- zur Wiederholung, Zusammenfassung und Systematisierung verwendet werden.

In 13/1 ist eines der vorgegebenen Wahlthemen verbindlich. Die Bearbeitung eines anderen Wahlthemas ist im Rahmen der im Abschnitt 2 getroffenen Regelungen möglich.

Die Hinweise zum Unterricht enthalten Anregungen zu inhaltlichen Schwerpunktsetzungen, Ausprägungen von Sach-, Methoden- und Sozialkompetenz, Abstimmungen zu anderen Fächern sowie zur Nutzung von Lehr- und Lernmitteln. Sie tragen ausdrücklich Empfehlungscharakter und stellen eine Planungshilfe dar. Durch die Auswahl aus den Hinweisen in Verbindung mit eigenen Vorstellungen und Erfahrungen der Lehrkraft kann der konkreten Klassen- und Unterrichtssituation Rechnung getragen werden.

4 Grundsätze der Unterrichtsgestaltung

4.1 Didaktische Grundsätze

Im Interesse einer schülerorientierten Unterrichtsgestaltung greift der Informatikunterricht vorhandene Erfahrungen und Vorstellungen (auch Fehlvorstellungen) auf. Ausgehend davon soll eine Korrektur, Erweiterung, Vertiefung und Systematisierung von Kenntnissen, Fähigkeiten und Fertigkeiten zu den Grundlagen der Informatik, verbunden mit aktuellen Erkenntnissen und Forschungsvorhaben erreicht werden. Besonderer Wert wird auf die Ausprägung von praktischen Fähigkeiten und Fertigkeiten gelegt – als Ausdruck des handlungsorientierten Ansatzes dieser Rahmenrichtlinien. Dabei werden die Schülerinnen und Schüler insbesondere in der Projektphase an eine zunehmend selbstständige Arbeitsweise herangeführt. Hierzu ist es erforderlich, dass jedem Lernenden in der praktischen Arbeit ein Computerarbeitsplatz zur Verfügung steht.

Im Umgang mit Informatiksystemen spielen Hilfesysteme und ähnliche elektronische Nachschlagewerke eine zentrale Rolle. Aus diesem Grund sollten die Schülerinnen und Schüler die hierfür notwendigen Techniken erlernen und bewusst im Verlauf des gesamten Grundkurses anwenden. Insbesondere sei hier auch auf den fächerübergreifenden Aspekt dieser speziellen Nachschlagetechnik hingewiesen.

Ausgewählte Konzepte und Komplexitätsbetrachtungen fördern die Fähigkeit der Schülerinnen und Schüler zur Bewertung von Aufgabenlösungen.

Das Aufzeigen der Struktur eines Problems durch Modellierung führt zum geeigneten Lösungsmodell. In das informatische Modellieren wird mit den Phasen Problemgewinnung, informelle Problembeschreibung, formale Modellierung, Realisierung von Lösungsansätzen und Bewertung eingeführt. Verschiedene Modellierungsverfahren führen dabei zu verschiedenen Problemlösestrategien.

Komplexere Fragestellungen und Probleme und die in diesem Zusammenhang zu entwickelnden Lösungsstrategien können durch fächerübergreifendes Herangehen vermittelt werden. Die Projektphasen eignen sich zur Verbindung mit den Inhalten anderer Fächer (z. B. Wissensdatenbanken). Somit besteht die Möglichkeit, komplexer Probleme fächerübergreifend zu bearbeiten.

4.2 Leistungen und ihre Bewertung

Leistungsbewertungen geben den Schülerinnen und Schülern sowie den Erziehungsberechtigten Auskunft über den Leistungsstand und der Lehrkraft die Möglichkeit zu einem Vergleich zwischen den angestrebten und den erreichten Lernzielen. Leistungsnachweise sind die Grundlage einer Leistungsbewertung. Sie umfassen mündliche und schriftliche Nachweise sowie praktische Arbeiten am Computer und die Projektarbeit. Die Fach-, Methoden-, Sozial- und Selbstkompetenzen der Schülerinnen und Schüler sind beim Leistungsnachweis und der Leistungsbewertung in angemessener Weise zu berücksichtigen. Über Formen und Anzahl von Leistungsnachweisen entscheidet die Lehrkraft eigenverantwortlich auf der Grundlage der gültigen Erlasse, sowie der Beschlüsse der Fach- und Gesamtkonferenzen der Schule.

5 Inhalte

5.1 Übersichten

5.1.1 Themen und Kurse in der Sekundarstufe II

Schuljahrgang 11 (Einführungsphase)

Thema:	Zeitrichtwert
Projektarbeit unter Nutzung von Standardsoftware	14 Std.
Grundlagen der Informationstechnik	16 Std.
Datenschutz, Datensicherheit und Softwarerecht	5 Std.
Computer-Netzwerke	25 Std.

Schuljahrgänge 12/13 (Qualifikationsphase)

12/1

Thema:	Zeitrichtwert
Informatik und Gesellschaft	6 Std.
Algorithmenstrukturen und ihre Implementierung	24 Std.

12/2

Thema:	Zeitrichtwert
Datenstrukturen	10 Std.
Iteration und Rekursion	5 Std.
Informatisches Modellieren	15 Std.

13/1

Wahlthemen:	Zeitrichtwert
1. Modellbildung und Simulation	30 Std.
2. Analyse und Design eines Informatiksystems	30 Std.
3. Computergrafik	30 Std.
4. Abstrakte Datentypen und ihre Implementierung	30 Std.
5. Endliche Automaten und formale Sprachen	30 Std.
6. Kryptologie	30 Std.
7. Datenbankanwendungen zur dynamischen Webseiten-generierung	30 Std.

13/2

Thema:	Zeitrichtwert
--------	---------------

Projektarbeit	15 Std.
---------------	---------

5.2 Darstellung der Themen in der Sekundarstufe II

5.2.1 Schuljahrgang 11 (Einführungsphase)

Thema: Projektarbeit unter Nutzung von Standardsoftware

ZRW: 14 Std.

Vorbemerkungen/Qualifikationen

Die im Rahmen der fächerübergreifenden IKG gewonnenen Kenntnisse, Fähigkeiten und Fertigkeiten im Umgang mit Standardsoftware sollen weiter gefestigt und aus informatischer Sicht systematisiert werden.

Die Schülerinnen und Schüler

- kennen wesentliche Grundfunktionen von Standardsoftware und können diese auf verschiedene Softwareprodukte anwenden,
- können produktbezogene Hilfesysteme zur Lösung ihrer Aufgaben nutzen,
- können mit geeigneter Standardsoftware thematische Belegarbeiten/Präsentationen anfertigen,
- kennen Objekte innerhalb von Standardsoftware,
- kennen Eigenschaften dieser Objekte und können diese verändern.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Grundfunktionen von Standardsoftware <ul style="list-style-type: none"> • Dateiarbeit und Dokumentenverwaltung • Formatierung von Objekten – Nutzung von produktbezogenen Hilfesystemen – Erstellen einer thematischen Belegarbeit/Präsentation unter Verwendung mehrerer Standardsoftwareprodukte 	<ul style="list-style-type: none"> – systematisches Herausarbeiten wesentlicher und übereinstimmender Grundfunktionen – Textobjekte (Zeichen, Word, Absatz, Seiten, Abschnitt, Dokument) – Externe Objekte (Grafik, Video, Sound, WordArt, ...) – die thematische Belegarbeit/Präsentation kann im Laufe des Schuljahres angefertigt und in entsprechender Form verteidigt und bewertet werden – hierbei sind nicht nur informatische Themen zulässig, sondern es ist auch das fächerübergreifende Arbeiten zu

	ermöglichen
--	-------------

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none">– objektorientierte Betrachtungsweisen von Standardsoftware<ul style="list-style-type: none">• Begriffe: Klasse, Objekt, Methode, Eigenschaft (Attribut)• Identifikation der Objekte• Eigenschaften dieser Objekte• Analyse der Methoden auf Objekte	<ul style="list-style-type: none">– die Behandlung hat an mehreren Objekten zu erfolgen– Beispiel:<ul style="list-style-type: none">• Klasse: WordArt-Grafik• Objekt: konkreter WordArt-Schriftzug im Text• Methoden: kopieren, verschieben, Methoden des Kontextmenüs• Eigenschaften: Größe, Farbe, Position

Thema: Grundlagen der Informationstechnik**ZRW: 16 Std.****Vorbemerkungen/Qualifikationen**

Schwerpunkt dieses Themas ist die Vermittlung eines grundlegenden Verständnisses über den Aufbau und die Funktionsweise eines Computersystems. Es werden elementare Fähigkeiten und Fertigkeiten aus dem Bereich der Technischen Informatik herausgebildet. Die eigentlichen technischen Komponenten werden hierbei als funktionale Einheit (Black Box) benutzt und nur Wert auf ihre Logik gelegt. Aufbauend auf den neu gewonnenen technischen Kenntnissen über die Funktionsweise von Halb- und Volladdierer wird die grundlegende Programmierung einfacher Operationen zum Beispiel der Addition und der Multiplikation ganzer Zahlen auf Maschinenebene (Assembler) vorgenommen. Basierend auf diesen Kenntnissen wird anschließend das Betriebssystem als zentrale Software des Rechners behandelt. Hierbei werden zentrale Strukturen und Funktionen vermittelt, ebenso grundlegende Fertigkeiten und Fähigkeiten im Umgang mit Betriebssystemen.

Die Schülerinnen und Schüler

- kennen den elementaren Aufbau und die prinzipielle Arbeitsweise einer Rechenanlage nach J. v. Neumann,
- kennen elementare Schaltfunktionen (Grundgatter) als elementare Bestandteile der Baugruppen eines J. v. Neumann-Rechners,
- können einfache logische Schaltnetze aus Grundgattern entwerfen und analysieren,
- können die Funktionsweise eines Halbaddierers, Volladdierers und FlipFlops erklären,
- kennen die Semantik und die Syntax einer Assemblersprache im benötigten Umfang (Modellrechner oder realer Rechner),
- können einfache Assemblerprogramme analysieren und selber entwickeln,
- kennen die Aufgaben von Betriebssystemen und sind in der Lage, mittels geeigneter Diagnose- und Verwaltungssoftware laufende Prozesse des Computers festzustellen und die Systemressourcen zu untersuchen,
- kennen das Schichtenmodell einer Rechneranlage und verstehen das Betriebssystem als Mittler zwischen Hard- und Software,
- können mittels eines geeigneten Dateimanagements Ordnerstrukturen und Dateien auf Datenträgern verwalten.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Aufbau eines Rechners nach J. v. Neumann <ul style="list-style-type: none"> • Rechenwerk • Steuerwerk • Speicherwerk • Ein- und Ausgabekanäle – Funktionsmerkmale des Rechenwerks, Steuerwerks und Speicherwerks und ihr prinzipielles Zusammenwirken 	<ul style="list-style-type: none"> – Ausblick auch auf andere Rechnerarchitekturen
<ul style="list-style-type: none"> – Begriffe Kodieren und Dekodieren <ul style="list-style-type: none"> • Umrechnung zwischen Zahlensystemen • ASCII/ANSI-Code 	<ul style="list-style-type: none"> – Wiederholung Dualzahlen – Binärsystem, Dezimalsystem, Hexadezimalsystem
<ul style="list-style-type: none"> – grundlegende logische Schaltfunktionen (Grundgatter) und ihre Schaltbelegungstabellen – Analyse und Entwurf einfacher Schaltetze bestehend aus Grundgattern 	<ul style="list-style-type: none"> – Simulation der Schaltfunktionen mittels Tabellenkalkulation oder mittels Simulationssoftware
<ul style="list-style-type: none"> – Umsetzung der Addition mit Hilfe logischer Schaltelemente <ul style="list-style-type: none"> • Halbaddierer mit Schaltbelegungstabelle • Volladdierer mit Schaltbelegungstabelle 	<ul style="list-style-type: none"> – Wiederholung Operationen mit binären Zahlen
<ul style="list-style-type: none"> – Speicherung eines Bits mit Hilfe logischer Schaltelemente – FlipFlop mit Schaltbelegungstabelle 	
<ul style="list-style-type: none"> – Beschreibung der Funktionsweise eines J. v. Neumann Rechners am Beispiel eines Modellrechners – Befehle eines Modellrechners in Assemblerschreibweise 	<ul style="list-style-type: none"> – oder an einem realen Rechner – Ein- und Ausgabebefehl – Transportbefehle – Arithmetische Befehle – Sprungbefehle
<ul style="list-style-type: none"> – Schrittfolge vom Assembler Quelltext bis zur Programmausführung <ul style="list-style-type: none"> • Lexikografische Prüfung • Syntaxprüfung • Befehlsinterpretation 	

- | | |
|---------------------|--|
| • Befehlsausführung | |
|---------------------|--|

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Lösen einfacher Aufgaben mit dem genutzten Modellrechner 	<ul style="list-style-type: none"> – Addition zweier natürlicher Zahlen – Multiplikation zweier natürlicher Zahlen – weitere einfache Aufgaben
<ul style="list-style-type: none"> – Begriffsbestimmung Betriebssystem – Aufgaben von Betriebssystemen <ul style="list-style-type: none"> • Abstraktion von Hardware • Verwaltung von Prozessen (z. B. Programme in Ausführung) und Betriebsmitteln (z. B. Arbeitsspeicher und periphere Geräte) 	<ul style="list-style-type: none"> – als Mittler zwischen Hardware und Software – nach Möglichkeit sollte dem Schüler mindestens ein weiteres Betriebssystem praktisch näher gebracht werden
<ul style="list-style-type: none"> – Einteilungsmöglichkeiten und Beispiele bezüglich: <ul style="list-style-type: none"> • User (Single-User/Multi-User) • Tasking (Single-Tasking/Multi-Tasking) 	
<ul style="list-style-type: none"> – Schichtenmodell (Schalenmodell) einer Rechenanlage <ul style="list-style-type: none"> • Anwendersoftware • Betriebssystem • ROM-Software • Hardware 	<ul style="list-style-type: none"> – Einordnung von Softwareprodukten in das Schichtenmodell – Beispiel: BIOS beim PC
<ul style="list-style-type: none"> – Schichten eines Betriebssystems 	<ul style="list-style-type: none"> – Kern, Treiber, Filesystem
<ul style="list-style-type: none"> – Umgang mit einem Dateimanagement – Arbeit mit Diagnose-, Verwaltungs- und Wartungsprogrammen 	<ul style="list-style-type: none"> – Beispiel: Windows-Explorer – Beispiele aus der Windows-Welt: Taskmanager, Taskplaner, Scandisk, Defrag, Systemmonitor ...

Thema: Datenschutz, Datensicherheit und Softwarerecht

ZRW: 5 Std.

Vorbemerkungen/Qualifikationen

In diesem Thema werden rechtliche und sicherheitsrelevante Fragestellungen behandelt, die sich aus der Nutzung von Informatiksystemen im soziotechnischen Umfeld ergeben.

Die Schülerinnen und Schüler

- sind mit der Problematik des Datenschutzes und der Datensicherheit vertraut,
- können zwischen Datenschutz und Datensicherheit unterscheiden,
- kennen wesentliche gesetzliche Grundlagen des Datenschutzes und
- können daraus Persönlichkeitsrechte und Pflichten ableiten,
- wissen um grundlegende strafrechtlichen Konsequenzen,
- kennen die Bedeutung der Datensicherheit und sind in der Lage, entsprechende Softwareprodukte zur Wahrung dieser Datensicherheit einzurichten und zu nutzen,
- erkennen die Notwendigkeit des Softwarerechts und kennen verschiedene lizenzrechtliche Modelle.

Inhalte	Hinweise zum Unterricht
– Begriffe: Datenschutz, Datensicherheit	
– Persönlichkeitsrecht und Datenschutz – Datenschutzgesetz	– aktuelle Beispiele aus den Medien – Datenschutz an der Schule
– Datensicherheit <ul style="list-style-type: none"> • Backup • Computerviren/ Antivirenprogramme 	– Haltbarkeit von Datenträgern – Infektionswege – Arten von Computerviren – Schutzmaßnahmen
– Datenschutz und Datensicherheit in Netzwerken	– Datenschutz: <ul style="list-style-type: none"> • Bestandsdaten • Nutzungsdaten • Abrechnungsdaten • Inhaltsdaten – Datensicherheit: <ul style="list-style-type: none"> • Zugangskontrolle • Zugriffsrechte (Freigaben und Berechtigungen) • Firewalls • Signaturen

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none">– Urheberrecht und Softwarerecht<ul style="list-style-type: none">• Autoren- und Verwertungsrecht• Urheberrecht von Softwareprodukten• Lizenzformen	<ul style="list-style-type: none">– beachten der Rechtslage beim Anfertigen von Schülerprojekten– Kauflizenzen– rechner-spezifische Lizenzierung– Public Domain– Shareware– Freeware

Thema: **Computer-Netzwerke**

ZRW: 25 Std.

Vorbemerkungen/Qualifikationen

Alle im Thema aufgeführten Inhalte sollten vorrangig aus der Sicht des in der Schule vorhandenen Netzwerkes und dessen Möglichkeiten behandelt und praktisch realisiert werden.

Die Schülerinnen und Schüler

- kennen wesentliche Abschnitte der historischen Entwicklung von Netzwerken, speziell des Internets,
- kennen Aufgaben von Netzwerken,
- kennen Einteilungsmöglichkeiten für Netzwerke und können das vorhandene schulische Netzwerk einordnen,
- wissen, was ein Protokoll aus informationstechnischer Sicht ist,
- können ausgehend von einem einfachen Schichtenmodell die Notwendigkeit von Protokollen erläutern,
- kennen wesentliche Netzwerkdienste und die zugehörigen Protokolle,
- kennen Gemeinsamkeiten und Unterschiede zwischen Intranet und Internet,
- können eigene Webseiten und ein eigenes Webprojekt planen, erstellen und auf einem Webserver implementieren.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – globale und lokale Netzwerke <ul style="list-style-type: none"> • Begriffsbestimmung • historische Aspekte 	
<ul style="list-style-type: none"> – Aufgaben von Netzwerken <ul style="list-style-type: none"> • Datenfernübertragung • gemeinsame Nutzung von Ressourcen und von Peripherie • Arbeitsteilung in Netzen 	<ul style="list-style-type: none"> – am Beispiel des eigenen Netzwerkes erläutern und bewusst nutzen
<ul style="list-style-type: none"> – Datensicherheit in Netzwerken 	<ul style="list-style-type: none"> – siehe Thema „Datenschutz, Datensicherheit und Softwarerecht“ – spezielle Hard- und Softwarelösungen zur Realisierung der Datensicherheit
<ul style="list-style-type: none"> – Kommunikation in Netzwerken <ul style="list-style-type: none"> • Paketvermittlung • Leitungsvermittlung 	

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Zugriffsverfahren <ul style="list-style-type: none"> • CSMA/CD (stochastisches Verfahren) • Token Passing (deterministisches Verfahren) 	
Einteilungsmöglichkeiten von Netzwerken <ul style="list-style-type: none"> – nach Aufgabenverteilung <ul style="list-style-type: none"> • Client-Server • Peer to Peer – nach Topologien <ul style="list-style-type: none"> • Bustopologie • Ringtopologie • Sterntopologie – nach Übertragungsweiten – nach Abgeschlossenheit <ul style="list-style-type: none"> • Intranet • Internet 	<ul style="list-style-type: none"> – Bezug zur Datensicherheit – die unterschiedlichen Sichtweisen auf Netzwerke sollten am Beispiel des eigenen Netzwerkes behandelt werden – LAN, MAN, WAN, GAN
<ul style="list-style-type: none"> – Protokolle <ul style="list-style-type: none"> • Begriffsbestimmung • Aufgaben • Zusammenhang zwischen Protokoll und Diensten 	<ul style="list-style-type: none"> – der Zusammenhang zwischen Protokoll und Diensten ist an geeigneten Beispielen aufzuzeigen
<ul style="list-style-type: none"> – einfaches Schichtenmodell <ul style="list-style-type: none"> • Applikationsschicht • Netzwerkkommunikationsprotokollschicht • Netzwerktransportprotokollschicht • Netzwerknormen 	
<ul style="list-style-type: none"> – Adressierung in TCP/IP-Netzwerken <ul style="list-style-type: none"> • IP-Adresse • DNS-Dienst (Domain Name Server) • feste Adressierung • dynamische Adressierung • Netzwerkklassen • Routing 	<ul style="list-style-type: none"> – Klasse A, B, C und D-Netze – DHCP-Server (Dynamic Host Configuration Protocol)
<ul style="list-style-type: none"> – Erstellung eines Webprojekts <ul style="list-style-type: none"> • Syntax und Semantik einer Seitenbeschreibungssprache (HTML, XHTML) • Nutzung eines geeigneten Editors • Integration clientseitiger Funktionalität 	<ul style="list-style-type: none"> – beim Einsatz visueller Editoren ist der erzeugte Quelltext zu interpretieren und zu bearbeiten – Auswertung von Formularen (JavaScript, VBScript, ...)

5.2.2 Schuljahrgänge 12/13 (Qualifikationsphase)

Schuljahrgang 12/1

Thema: Informatik und Gesellschaft

ZRW: 6 Std.

Vorbemerkungen/Qualifikationen

Erst durch die Kenntnis von Voraussetzungen und Folgen, Chancen und Risiken des Einsatzes komplexer Informatiksysteme werden Schülerinnen und Schüler in die Lage versetzt, sich verantwortungsbewusst an der Gestaltung und am Einsatz dieser Technologie zu beteiligen und ihre Zukunft menschengerecht zu gestalten. Dazu setzen sie sich auch mit normativen und ethischen Fragen auseinander, die zum Beispiel den Zugriff auf personenbezogene Daten oder den Umgang mit dem Urheberrecht betreffen. Aus der Kenntnis der Wirkungen des Einsatzes von Informatiksystemen auf Individuum und Gesellschaft heraus sollen sie Kriterien für menschengerechte Technikgestaltung und deren sozialverträglichen Einsatz entwickeln können. Überhöhten Erwartungen an das Machbare sollen sie ebenso entgegentreten wie fatalistischen Einstellungen des Ausgeliefertseins gegenüber Informatiksystemen.

Die Schülerinnen und Schüler

- kennen die Teilgebiete der Informatik und haben einen Einblick in ihre Aufgaben als Wissenschaft,
- kennen Persönlichkeiten und ihre Leistungen, sowie technische und theoretische Entwicklungen aus der Historie der Informatik,
- kennen Einsatzmöglichkeiten von Informatiksystemen,
- erkennen Informatiksysteme in ihrer täglichen Umwelt,
- kennen Voraussetzungen und Folgen, Chancen, Risiken und Grenzen des Einsatzes von Informatiksystemen.

Inhalte	Hinweise zum Unterricht
Informatik – Begriffsbestimmung – Teilgebiete <ul style="list-style-type: none"> • Technische Informatik • Theoretische Informatik 	

- | | |
|---|--|
| <ul style="list-style-type: none">• Angewandte Informatik• Praktische Informatik | |
|---|--|

Inhalte	Hinweise zum Unterricht
<p>Historie</p> <ul style="list-style-type: none">– Persönlichkeiten– (Leibniz, Babbage, Hollerith, Turing, Zuse, von Neumann, Chomsky)– Hardware- und Softwareentwicklungen<ul style="list-style-type: none">• Computergenerationen• Generationen von Softwareentwicklungswerkzeugen	<ul style="list-style-type: none">– Vorstellung der Persönlichkeiten und Würdigung ihrer Leistungen durch Schülervorträge
<ul style="list-style-type: none">– Einsatzgebiete von Informatiksystemen	<ul style="list-style-type: none">– Beispiele:<ul style="list-style-type: none">• elektronische Zahlungssysteme (Registrierkasse)• elektronische Zugangs-, Identifikations- und Arbeitszeiterfassungssysteme
<ul style="list-style-type: none">– Grenzen von Informatiksystemen<ul style="list-style-type: none">• Ethik• Datenschutz• Berechenbarkeit• Physikalische Ebene	

Thema: Algorithmenstrukturen und ihre Implementierung

ZRW: 24 Std.

Vorbemerkungen/Qualifikationen

In diesem Thema werden ausgehend vom Algorithmusbegriff Algorithmenstrukturen eingeführt und in einer Programmiersprache implementiert. Die gewählte Programmiersprache sollte den objektorientierten Softwareentwicklungsansatz unterstützen.

Die Schülerinnen und Schüler

- kennen einen Algorithmusbegriff und Eigenschaften von Algorithmen,
- können Algorithmen verbal formulieren und in Struktogrammen darstellen,
- können Algorithmen analysieren, interpretieren und in einer Programmiersprache codieren,
- sind in der Lage, Programmabschnitte zu strukturieren und zu entsprechenden Einheiten zusammenzufassen,
- können Programmoberflächen gestalten und die Eigenschaften der visuellen Komponenten bearbeiten und ihre Methoden (Ereignisse) mit entsprechendem Quelltext versehen,
- kennen grafische Basisroutinen der verwendeten Programmiersprache und können diese anwenden.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – intuitive Definition des Algorithmusbegriffs – Eigenschaften <ul style="list-style-type: none"> • Terminiertheit • Determiniertheit • Finitheit • Abstraktion 	<ul style="list-style-type: none"> – Handlungsvorschriften aus dem Alltag
<ul style="list-style-type: none"> – Beschreibungsmittel für Algorithmen <ul style="list-style-type: none"> • Verbale Formulierung • Struktogramm – Algorithmenstrukturen: <ul style="list-style-type: none"> • Sequenz • Verzweigung • Zyklen 	<ul style="list-style-type: none"> – die Beschreibungsmittel sind an geeigneten Beispielen vorzustellen – im gesamten Thema sind die zu behandelnden Algorithmenstrukturen mittels Struktogrammen darzustellen – zur Darstellung sollte ein geeigneter Struktogrammeditor genutzt werden

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none">– Analyse und Darstellung von einfachen Algorithmen in Struktogrammen<ul style="list-style-type: none">• allgemeine quadratische Gleichung mittels Lösungsformel• Wachstumsfunktionen	<ul style="list-style-type: none">– die Algorithmenstrukturen sollen in der verwendeten Programmiersprache implementiert werden– weitere Beispiele aus anderen Fächern
<p>Objektorientierte Softwareentwicklung</p> <ul style="list-style-type: none">– Datenmodell<ul style="list-style-type: none">• konkrete Datentypen– Visuelle Darstellung<ul style="list-style-type: none">• Ein- und Ausgabekonzept– Methoden/Algorithmenimplementierung<ul style="list-style-type: none">• Anweisungen, Projektstrukturierung	<ul style="list-style-type: none">– Nutzung von visuellen Entwicklungsumgebungen– Zugang über die vordefinierten visuellen Objektklassen
<ul style="list-style-type: none">– Grafische Basisroutinen<ul style="list-style-type: none">• Aufruf von Grafikroutinen des gewählten Grafikpaketes in der verwendeten Programmiersprache	<ul style="list-style-type: none">– Veranschaulichung von Daten in Diagrammen und Koordinatensystemen

Schuljahrgang 12/2**Thema: Datenstrukturen****ZRW: 10 Std.****Vorbemerkungen/Qualifikationen**

Im Rahmen dieses Themas werden Datentypen zur Verwaltung von komplexen Datenstrukturen eingeführt. Hierzu werden Methoden zur permanenten Verwaltung auf externen Datenträgern vermittelt.

Die Schülerinnen und Schüler

- kennen Strukturen zur Verwaltung komplexer Daten und zur rechnerinternen Repräsentation,
- können Daten in Dateien verschiedenen Typs verwalten.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Strukturierte Datentypen <ul style="list-style-type: none"> • Reihung (Felder) • Datenverbund – Implementierung der genannten Datentypen in der verwendeten Programmiersprache 	<ul style="list-style-type: none"> – Operationen auf Vektoren und Matrizen – Erstellen von Datensätzen aus einem Applikationsgebiet
<p>Dateiarbeit</p> <ul style="list-style-type: none"> – Dateitypen <ul style="list-style-type: none"> • typisierte Dateien • Textdateien – Operationen auf Dateien <ul style="list-style-type: none"> • Speichern und Öffnen von Dateien • Zugriffsverfahren auf Daten • Suchen, Editieren und Speichern von Daten • Fehlerinterpretation – Zusammenwirken von Betriebssystem und Dateizugriffskonzept 	<ul style="list-style-type: none"> – Verwaltung von Datensätzen aus einem Applikationsgebiet – (X)HTML-Seiten aus Datensätzen generieren

Thema: Iteration und Rekursion**ZRW: 5 Std.****Vorbemerkungen/Qualifikationen**

In diesem Abschnitt werden Iteration und Rekursion als fundamentale Verfahren zur Beschreibung von Zyklen behandelt.

Die Schülerinnen und Schüler

- kennen unterschiedliche Formen der Gestaltung von Zyklen,
- können iterative und rekursive Algorithmen erkennen und entwerfen,
- kennen unterschiedliche Methoden der Terminierung der Zyklenform.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none">– Iteration und Rekursion<ul style="list-style-type: none">• Vergleich beider Verfahren• Terminierungsbedingungen• Beispielalgorithmen entwerfen und implementieren	<ul style="list-style-type: none">– Vergleich der Verfahren an Hand der binären Suche– Fakultät, Fibonacci-Folgen, Turm von Hanoi

Thema: Informatisches Modellieren

ZRW: 15 Std.

Vorbemerkungen/Qualifikationen

Die im Umgang mit Algorithmierung und Programmiersprache erlernten Fähigkeiten und Fertigkeiten werden an einem komplexen Beispiel gefestigt. Es wird dabei ein vollständiger Softwareentwicklungszyklus (Software life cycle) durchlaufen. Die Gruppenarbeit sollte die bevorzugte Arbeitsform bei der Softwareentwicklung sein.

Die Schülerinnen und Schüler

- sind in der Lage, aus einer Aufgabenstellung heraus die Ziele und Anforderungen für ein Informatiksystem abzuleiten,
- erkennen das Informatiksystem als Abbild (Modell) eines realen Systems,
- können die erforderlichen Daten bestimmen und in der geforderten Genauigkeit erfassen,
- können unter Anleitung ein komplexes Problem in Teilprobleme zerlegen und die Funktionen dieser Bausteine und ihre Schnittstellen implementieren sowie die Teillösungen zu einer Gesamtlösung zusammenfassen (Top-down-Methode und Bottom-Up-Methode),
- kennen die Bedeutung von Verifikation und Validierung für den Prozess der Modellbildung und der Softwareentwicklung,
- sind in der Lage, eine projektbegleitende Dokumentation zu erstellen,
- begreifen den Softwarelebenszyklus als iterativen Prozess und erkennen dessen wirtschaftliche Bedeutung.

Inhalte	Hinweise zum Unterricht
Phasen des Softwareentwicklungszyklus – Problemformulierung und Problemanalyse <ul style="list-style-type: none"> • Ergebnis: Anforderungsdefinition 	– vollständiges und eindeutiges Erfassen des realen Systems bzw. Problems und aller wichtigen Umgebungsbedingungen (Systemeigenschaften, Pflichtenheft)
– Entwurf <ul style="list-style-type: none"> • Ergebnis: Modellbildung • Ergebnis: Spezifikation 	– Beschreibung des zu erzeugenden Systems mit informalen und formalen Mitteln und Erzeugung eines abstrakten Modells – Zerlegung des komplexen Gesamtsystems in überschaubare Teilprobleme (Module) und ihre

	Bearbeitung in Gruppen
--	------------------------

Inhalte	Hinweise zum Unterricht
	<ul style="list-style-type: none"> – Zusammenführen dieser Teilprobleme durch Beschreibung der Modulfunktionen und einer eindeutigen Schnittstellenbeschreibung sowie aufzeigen vorhandener Modulabhängigkeiten
<ul style="list-style-type: none"> – Implementierung <ul style="list-style-type: none"> • Ergebnis: lauffähiges dokumentiertes Programm 	<ul style="list-style-type: none"> – Programm entspricht dem Ein-/Ausgabeverhalten der Anforderungsdefinition
<ul style="list-style-type: none"> – Funktions- und Leistungsüberprüfung <ul style="list-style-type: none"> • Ergebnis: modifiziertes Programm 	<ul style="list-style-type: none"> – Verifikation und Testen der Einzelmodule – Integrationstest – Installationstest – Laufzeit- und Speicherverhalten
<ul style="list-style-type: none"> – Installation und Abnahme <ul style="list-style-type: none"> • Ergebnis: fertiges Softwareprodukt entsprechend der Anforderungsdefinition 	<ul style="list-style-type: none"> – Integration des Softwareproduktes in die Systemumgebung des Auftraggebers
<ul style="list-style-type: none"> – Wartung – Modellkritik <ul style="list-style-type: none"> • Validierungstechniken • Ergebnisanalyse 	<ul style="list-style-type: none"> – Anpassen der Software an neue Anforderungen – Implementierung effizienterer Algorithmen – Beseitigung noch existierender Mängel
<ul style="list-style-type: none"> – Dokumentation 	<ul style="list-style-type: none"> – die Dokumentation ist über den gesamten Zeitraum dieser Unterrichtseinheit zu führen

Schuljahrgang 13/1**Wahlthema 1: Modellbildung und Simulation****ZRW: 30 Std.****Vorbemerkungen/Qualifikationen**

Im Rahmen dieses Themas werden die Grundlagen der Modellbildung und Simulation vermittelt. Aufbauend auf den in der Qualifizierungsphase gewonnenen Erkenntnissen aus dem Bereich der Softwareentwicklung werden diese am Beispiel der Entwicklung von Simulationsstudien gefestigt.

Die Schülerinnen und Schüler

- können geeignete Abstraktionstechniken zur Erstellung von Simulationsmodellen anwenden,
- erkennen, dass ein Modell einen Ausschnitt aus der realen Welt entsprechend des gewählten Abstraktionsniveaus beschreibt,
- erfahren den Erkenntnisgewinn durch Analogieschlussverfahren am Beispiel von Simulationsexperimenten mit diesen Modellen,
- erlernen den Umgang mit einer Simulationssprache bzw. einem Simulationswerkzeug und sind in der Lage, einfache Computermodelle selbst zu implementieren,
- erlernen Methoden der Visualisierung von Resultaten,
- sind in der Lage, Resultate eines Simulationslaufes und einer Simulationsstudie verbal zu interpretieren und kritisch zu bewerten.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Grundbegriffe <ul style="list-style-type: none"> • Modellbegriff • Simulationsmodell • reales System 	<ul style="list-style-type: none"> – Klassifikation von Modellen nach unterschiedlichen Kriterien
<ul style="list-style-type: none"> – Bediensystem als Simulationsmodell <ul style="list-style-type: none"> • Auswahl und Beschreibung eines Bediensystems • Definition des Systembegriffs • Methoden von Abstraktion und Reduktion • Erfassung und Aufbereitung empirischer Daten 	<ul style="list-style-type: none"> – Auswahl eines Bediensystems aus dem Erfahrungshorizont der Schüler, zum Beispiel Dienstleistungsbetrieb, Supermarkt, Tankstelle – Wachstumsmodelle als Beispiel für kontinuierliche Modelle – verbale Beschreibung des realen Systems – Auftrag zur Datenerhebung – Definition des Ziels der Simulations-

	studie, zum Beispiel Untersuchung von Strategien
--	--

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> • Rollenspiel des zu untersuchenden Systems • formale Ablaufbeschreibung • Simulation als Prozess • Zeitraffung und Zeitstreckung im Modell • Begriff der Validierung 	<ul style="list-style-type: none"> – Entwicklung und Vertiefung des Systemverständnisses – Validierung des abstrakten Modells – Erzeugen einer Ablaufskizze
<ul style="list-style-type: none"> – Modellimplementierung <ul style="list-style-type: none"> • Anweisungen, Datentypen und Abarbeitungsstrukturen einer Simulationssprache • Abbildung zeitparalleler Abläufe • Verwaltung der Simulationsuhr • Implementierung des gewählten Modells in der Simulationssprache • Verifikation und Validierung des Computermodells 	
<ul style="list-style-type: none"> – Simulationsexperimente <ul style="list-style-type: none"> • Pseudozufallszahlengeneratoren • Interpretation abstrakter Resultate • Simulation als iterativer Prozess 	<ul style="list-style-type: none"> – Ableitung von systematischen Experimentierstrategien – Durchführen von Experimentserien mit unterschiedlichen Zufallszahlen
<ul style="list-style-type: none"> – Modifikation des Modells <ul style="list-style-type: none"> • Vertiefen der Anweisungen der Simulationssprache • Modifikation des Modells 	<ul style="list-style-type: none"> – Änderung der Priorität von Agenten (Forderungen) – Anwendung verschiedener Zufallszahlengeneratoren – Einführung erweiternder Sprach-elemente
<ul style="list-style-type: none"> – Visualisierung der Resultate <ul style="list-style-type: none"> • Visualisierungs- und Präsentationstechniken • Nutzung von Schnittstellen zur Standardsoftware 	<ul style="list-style-type: none"> – Aufbereitung und Bewertung der Resultatdaten – Anwendung von Standardsoftware zur Präsentation – Nutzung von Animationssystemen
<ul style="list-style-type: none"> – Modellkritik <ul style="list-style-type: none"> • Validierungstechniken • Ergebnisanalyse 	<ul style="list-style-type: none"> – Abschließende Betrachtung und Wertung des Modellbildungsprozesses – kritische Betrachtung des Modellbildungsprozesses – Simulationsresultate als Hilfsmittel zur Entscheidungsunterstützung anwenden

Wahlthema 2: Analyse und Design eines Informatiksystems**ZRW: 30 Std.****Vorbemerkungen/Qualifikationen**

In diesem Thema werden wesentliche Grundkenntnisse im Bereich der Datenerfassung, der Datenaufbereitung, der Datenverarbeitung und der Steuerung eines Informatiksystems an Hand eines Modells vermittelt. Das gewählte Modell sollte sich an einem realen System aus der Umwelt orientieren. Hierfür geeignet sind funktionale Modelle aus dem Bereich der Mess-, Steuerungs- und Regelungstechnik.

Die Schülerinnen und Schüler

- kennen Möglichkeiten der Datenerfassung und Datenaufbereitung bezogen auf das gewählte Modell,
- gelangen, ausgehend von der Problemstellung über die Modellierung des Systems zu problemadäquaten Datenstrukturen und Algorithmen,
- können diese in einer hierfür geeigneten Sprache implementieren,
- verstehen das Modell als eine Einheit aus mechanischer Funktionalität, Hard- und Software,
- kennen Sensoren und können deren Signale erfassen und auswerten,
- kennen Aktoren und können diese zur Modellsteuerung nutzen,
- können das gewählte Modell steuern.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Automatisierung und Einsatz von Robotern in der Industrie und die damit verbundenen gesellschaftlichen Auswirkungen 	<ul style="list-style-type: none"> – Interdisziplinäre Aspekte (Arbeits- und Sozialwissenschaften, Maschinenbau, Elektro- und Kommunikationstechnik) – fachübergreifende Aspekte (physikalische Wirkprinzipien)
<ul style="list-style-type: none"> – Steuerung – Regelung 	<ul style="list-style-type: none"> – Darstellen und Erläutern einer Steuerkette – Darstellen und Erläutern eines Regelkreises
<ul style="list-style-type: none"> – Elementare Algorithmen zum Steuern und Regeln und ihre Implementierung 	<ul style="list-style-type: none"> – Kontrollstrukturen – Steuerungsalgorithmen
<ul style="list-style-type: none"> – Digitalisierung analoger Größen sowie die Umwandlung digitaler Signale in analoge 	<ul style="list-style-type: none"> – AD- und DA-Wandler – Skalierung analoger Messwerte – Vergleich der Effizienz von Steuerungs-

	algorithmen
Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Messwernerfassung und -verarbeitung <ul style="list-style-type: none"> • Abfrage von Sensoren – Prozessbeeinflussung <ul style="list-style-type: none"> • Ansteuern von Aktoren 	<ul style="list-style-type: none"> – Einfluss von Aktoren auf die Gerätetechnik – Kopplungen (galvanisch, optisch, magnetisch)
<ul style="list-style-type: none"> – Modellierung einfacher Steuerungsprozesse und ihre Implementierung 	
<ul style="list-style-type: none"> – Gestaltung von Benutzungssinterfaces 	<ul style="list-style-type: none"> – Funktionalität, Anwenderfreundlichkeit, Robustheit („Absturzsicherheit“) von Bedienoberflächen
<ul style="list-style-type: none"> – Einblick in objektorientierte Arbeitsweisen 	<ul style="list-style-type: none"> – Entwurf von Klassen und Methoden – Wiederverwendbarkeit der Software – Zerlegung der Aufgabe in Teilprobleme (Softwareentwicklung in der Gruppe)

Wahlthema 3: Computergrafik**ZRW: 30 Std.****Vorbemerkungen/Qualifikationen**

Die Visualisierung von Daten und Prozessen gewinnt immer mehr an Bedeutung. In diesem Wahlthema soll, ausgehend von technischen und mathematischen Grundlagen, das prinzipielle Vorgehen beim Erstellen von virtuellen Welten an geeigneten Beispielen nachempfunden werden.

Die Schülerinnen und Schüler

- kennen Anwendungsgebiete der Computergrafik,
- kennen Basisalgorithmen der Computergrafik und können diese in einer geeigneten Programmiersprache implementieren,
- kennen ein additives und ein subtraktives Farbmodell und ihre Anwendungsgebiete,
- können Pixel- und Vektorgrafiken unterscheiden,
- kennen Transformationsarten und können diese implementieren,
- können einfache virtuelle Welten modellieren und diese mittels eines geeigneten Grafiksystems implementieren.

Inhalte	Hinweise zum Unterricht
Computergrafik <ul style="list-style-type: none"> – geschichtliche Entwicklung – Anwendungsgebiete <ul style="list-style-type: none"> • Bilderzeugung • Bildverarbeitung 	<ul style="list-style-type: none"> – Beispiele für Bilderzeugungssysteme <ul style="list-style-type: none"> • CAD-Systeme • Computerspiele • Schriftzeichen – Beispiele für Bildverarbeitungssysteme <ul style="list-style-type: none"> • Bildverbesserung • Mustererkennung • Szenenerkennung • optische Bilderfassung
Rastergrafik und Vektorgrafik <ul style="list-style-type: none"> – Begriffe – Vergleich von Raster- und Vektorgrafik – Gerätetechnik für Ein- und Ausgabe von Raster- bzw. Vektorgrafik 	

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Grafikdateiformate <ul style="list-style-type: none"> • Rastergrafikformate (BMP, TIF, GIF, PNG, JPEG) • Vektorgrafikformate (CDR, DXF) • Metafileformate (WMF) 	<ul style="list-style-type: none"> – BMP – Bitmap – GIF – Graphics Interchange Format – TIF – Tagget Image File – PNG – Portable Network Graphics – JPEG – Joint Photographic Experts Group – CDR – CorelDraw – DXF – Drawing Interchange Format (AutoCad) – WMF – Windows Meta File
<p>Rastergrafik</p> <ul style="list-style-type: none"> – Notwendigkeit von Basisalgorithmen – Rasterkonvertierungsalgorithmen <ul style="list-style-type: none"> • Mittelpunkt-Linien-Algorithmus nach Bresenham – Algorithmus zum Füllen von Rechtecken – Clipping <ul style="list-style-type: none"> • Cohen-Sutherland-Algorithmus – Implementierung der Basisalgorithmen in der gewählten Programmiersprache 	<ul style="list-style-type: none"> – Kompressionsverfahren – alternativ Bresenham Kreisalgorithmus
<ul style="list-style-type: none"> – Farbmodelle <ul style="list-style-type: none"> • Additives Farbmodell – RGB • Subtraktives Farbmodell – CMYK • Anwendungsgebiete und Umwandlung 	
<ul style="list-style-type: none"> – 2D-Transformationen/ 3D-Transformationen <ul style="list-style-type: none"> • Translation • Skalierung • Rotation • Komposition • Implementierung in der gewählten Programmiersprache 	<ul style="list-style-type: none"> – Verschiebung – Dehnung / Stauchung – Drehung – Nacheinanderausführung
<ul style="list-style-type: none"> – Modellierung und Darstellung (Rendering) virtueller Welten – Behandlung einer geeigneten Sprache 	<ul style="list-style-type: none"> – ist in dem Umfang zu behandeln, wie es für das Projekt in 13/2 notwendig ist – VRML, OpenGL, Direkt3D, Java3D, ...

Wahlthema 4: Abstrakte Datentypen und ihre Implementierung

ZRW: 30 Std.

Vorbemerkungen/Qualifikationen

In diesem Thema werden ausgewählte Datentypen ausführlich behandelt und Basisalgorithmen der Informatik analysiert. Zur formalen Beschreibung werden abstrakte Datentypen verwendet. Die Implementierung erfolgt mit einer geeigneten Programmiersprache.

Die Schülerinnen und Schüler

- kennen den Begriff des abstrakten Datentyps,
- können auf der Basis von abstrakten Datentypen Grundfunktionen auf die dynamische Datenstruktur Liste beschreiben und implementieren,
- kennen die Spezialformen Stapel und Warteschlange,
- kennen elementare und rekursive Sortieralgorithmen, können diese implementieren und bezüglich ihrer Zeitkomplexität bewerten,
- kennen Suchalgorithmen für typisierte Dateien und Textdateien und können diese implementieren.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Begriff und Eigenschaften des abstrakten Datentyp (ADT) 	<ul style="list-style-type: none"> – Universalität – Geheimnisprinzip
<ul style="list-style-type: none"> – Prinzipien der Verkettung der ADT-Liste <ul style="list-style-type: none"> • einfach verkettete Liste • doppelt verkettete Liste • Ringliste 	<ul style="list-style-type: none"> – Implementierung durch dynamische Datenstrukturen oder Reihung (Felder)
<ul style="list-style-type: none"> – Implementierung der Grundfunktionen – Methoden auf Listen <ul style="list-style-type: none"> • Initialisierung • Einketten • Ausketten • Suchen 	
<ul style="list-style-type: none"> – Anwendung von Listen <ul style="list-style-type: none"> • Stack (Stapel) • Queue (Warteschlange) 	<ul style="list-style-type: none"> – LIFO-Prinzip – FIFO-Prinzip – Nachbildung eines

	Bediensystemmodells
--	---------------------

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none">– Elementare Sortieralgorithmen<ul style="list-style-type: none">• Selectionsort• Insertionsort• Bubblesort	<ul style="list-style-type: none">– Sortieren von Feldern mit Zahlen– Sortieren von Feldern mit Datensätzen– Sortieren von Feldern nach verschiedenen Elementen des Datensatzes– Sortieren von Listen oder Dateien
<ul style="list-style-type: none">– Divide and Conquer Sortierverfahren<ul style="list-style-type: none">• Quicksort	
<ul style="list-style-type: none">– Suchverfahren<ul style="list-style-type: none">• sequentielles Suchen in Listen und Textdateien• iteratives und rekursives Suchen am Beispiel des binären Suchens	<ul style="list-style-type: none">– Listen oder dynamische Reihung (Felder)– siehe Thema „Iteration und Rekursion“
<ul style="list-style-type: none">– Zeitkomplexität von Algorithmen am Beispiel der zuvor behandelten Algorithmen als eine Möglichkeit der Bewertung von Algorithmen– Grenzen der Berechenbarkeit von Algorithmen	<ul style="list-style-type: none">– Abschätzen der Zeitkomplexität– Messen der benötigten Zeit– siehe Thema „Informatik und Gesellschaft“

Wahlthema 5: Endliche Automaten und formale Sprachen**ZRW: 30 Std.****Vorbemerkungen/Qualifikationen**

Ausgehend von der Arbeitsweise eines realen Automaten aus der Erfahrungswelt der Schülerinnen und Schüler und dessen abstrakter Beschreibung sind Modelle endlicher Automaten zu beschreiben und zu untersuchen sowie formale Sprachen zu behandeln.

Die Schülerinnen und Schüler

- kennen Einsatzfelder für Automaten aus ihrer Erfahrungswelt,
- können einen endlichen Automaten definieren,
- können endliche Automaten mit unterschiedlichen Beschreibungsmitteln beschreiben,
- kennen die Eigenschaften von speziellen endlichen Automaten (Akzeptor/Parser),
- wissen, was eine formale Sprache ist,
- können den Inhalt der Begriffe Terminalsymbol, Nichtterminalsymbol, Alphabet, Wort, Grammatik, Syntax, Syntaxdiagramme, Scanner, Parser und Interpreter im Zusammenhang mit formalen Sprachen erläutern,
- können Worte W_i einer formalen Sprache $L(G)$ – aus der Kenntnis ihrer Grammatik G – erzeugen oder entscheiden, ob ein Wort W aus Terminalsymbolen eines Alphabetes T zu einer vorgegebenen Sprache $L(G)$ gehört oder nicht,
- können Sprachen $L(G, \text{Syntax})$ zur Beschreibung von einfache Aufgaben entwickeln und deren Syntax sowie Semantik beschreiben.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Automaten <ul style="list-style-type: none"> • Automatenbegriff • Einsatzfelder von Automaten • Beispiel für einen endlichen Automaten • Grundmodell von Automaten 	<ul style="list-style-type: none"> – Der Automatenbegriff wird aus der Behandlung realer Automaten durch Modellbildung entwickelt
<ul style="list-style-type: none"> – Modell endlicher Automat <ul style="list-style-type: none"> • Definition endlicher Automat (Eingabemenge, Ausgabemenge, Menge der definierten Automatenzustände, Ausgabefunktion und Zustandsüberföhrungsfunktion) • Beschreibung endlicher Automaten 	<ul style="list-style-type: none"> – Beschreibungsmittel für das Modell endlicher Automat <ul style="list-style-type: none"> • formale Beschreibung wesentlicher Modellparameter • Automatentafeln

- | | |
|---|--|
| <ul style="list-style-type: none">• Simulation endlicher Automaten mit Computerprogrammen | |
|---|--|

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Automaten ohne Ausgabe als erkennende Automaten – Definition der besonderen Zustände als Fehlerzustand oder Zielzustand 	<ul style="list-style-type: none"> – Automaten nehmen besondere Zustände ein, wenn eine bestimmte Folge von Eingaben (Wort) erkannt und akzeptiert wird
<ul style="list-style-type: none"> – Parser als spezielle erkennende Automaten 	<ul style="list-style-type: none"> – Parser als endliche Automaten, bei denen die Folge von Eingabezeichen Worte sind
<ul style="list-style-type: none"> – Formale Sprachen <ul style="list-style-type: none"> • Begriffe und Definitionen (Sprache $L(G)$, Grammatik G, Nichtterminal, Terminalsymbol, Alphabet T als Menge der Terminalsymbole) – Erzeugen eines Wortes W bei gegebener Grammatik G – Nachweis, ob vorgegebene Worte zur Sprache $L(G)$ gehören 	<ul style="list-style-type: none"> – Zur Beschreibung einer (formalen) Sprache werden neben der abstrakten Beschreibung vor allem Syntaxdiagramme eingesetzt
<ul style="list-style-type: none"> – Entwurf und Definition (Syntax und Semantik) einer nichtformalen Sprache $L(G, \text{Syntax})$ 	<ul style="list-style-type: none"> – Bei den nichtformalen Sprachen werden den Worten W_i der Sprache Bedeutungen zugeordnet und es gilt: Wenn eine Aufgabe mit Hilfe einer Sprache $L(G, \text{Syntax})$ vollständig beschrieben werden kann, dann ist die Aufgabe prinzipiell gelöst
<ul style="list-style-type: none"> – Lösung von Aufgaben durch die Beschreibung dieser Aufgaben in einer selbstentwickelten Programmiersprache 	<ul style="list-style-type: none"> – Einsatz von nichtformalen Sprachen – auch selbstdefinierter – zur Lösung einfacher Aufgaben <ul style="list-style-type: none"> • Sprache zur Programmierung einfacher Assemblerprogramme • Sprache zur Umsetzung graphischer Basisalgorithmen

Wahlthema 6: Kryptologie**ZRW: 30 Std.****Vorbemerkungen/Qualifikationen**

In diesem Thema wird ausgehend von historischen Beispielen die gesellschaftliche Bedeutung und Notwendigkeit kryptologischer Verfahren dargestellt. Voraussetzung hierzu sind die Fähigkeiten und Fertigkeiten im Umgang mit der verwendeten Programmiersprache. Bei der mathematischen Beschreibung und Formalisierung sind bei einigen Verfahren Grenzen durch die Schulmathematik gesetzt.

Die Schülerinnen und Schüler

- kennen Beispiele aus der historischen Entwicklung und wissen um moderne gesellschaftliche Aspekte der Kryptologie,
- kennen wesentliche Aufgaben der Kryptologie in der Vergangenheit und Gegenwart,
- kennen verschiedene Verfahren der Kryptographie und der Kryptoanalyse,
- gelangen, ausgehend von der Problemstellung, zu adäquaten Algorithmen,
- können diese in eine hierfür geeigneten Programmiersprache implementieren,
- kennen Vor- und Nachteile ausgewählter Chiffrierverfahren,
- können Klartexte in der gewählten Programmiersprache nach verschiedenen Algorithmen chiffrieren,
- können fremde, verschlüsselte Texte analysieren und in der verwendeten Programmiersprache dechiffrieren,
- erkennen den starken interdisziplinären Charakter zwischen Informatik und Mathematik.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Grundlagen <ul style="list-style-type: none"> • Begriffsbestimmung und Bestandteile (Kryptographie, Kryptoanalyse, Steganographie) der Kryptologie • Entwicklung der Kryptologie 	<ul style="list-style-type: none"> – Aufzeigen wichtiger Meilensteine der Kryptologie und deren grundlegender Wirkprinzipien Beispiele: Skytale von Sparta, Cäsar-Scheibe, Chiffrierzylinder und Rotormaschinen, ENIGMA, Chip- und EC-Karten

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Gesellschaftliche und technische Notwendigkeit der Verschlüsselung von Daten <ul style="list-style-type: none"> • Datenschutzgesetze und ihre Geltungsbereiche • Grundprobleme von Datenschutz und Datensicherheit in offenen und verteilten Informatiksystemen • Sicherheitsanforderungen in offenen und verteilten Informatiksystemen 	<ul style="list-style-type: none"> – Ausgehend von dem im Schuljahrgang 11 erarbeiteten Wissen und eigenen Erfahrungen der Lernenden bieten sich, untersetzt mit Beispielen, folgende kryptographische Schutzmaßnahmen an: <ul style="list-style-type: none"> • Sicherung der Anonymität bei der Datenübertragung • Verschlüsselte Kommunikation • Digitale Signaturen • Authentisierung von Kommunikationspartnern • Sicherung der Integrität und der Verbindlichkeit
<p>Grundlagen der kryptographischen Theorie</p> <ul style="list-style-type: none"> – Algorithmen und Schlüssel – Funktionsprinzipien von Chiffrieralgorithmen <ul style="list-style-type: none"> • Monoalphabetische Chiffrierung und Polyalphabetische Chiffrierung • Transpositionschiffren, Substitutionschiffren und deren Kombination • symmetrische und asymmetrische Algorithmen • Bedeutung der Schlüssel – Kryptoanalyse <ul style="list-style-type: none"> • Sicherheit von Chiffrieralgorithmen • legale Kryptoanalyse und deren Grenzen – Verbergen von Informationen mit Hilfe der Steganographie 	<ul style="list-style-type: none"> – Hilfsmittel zur Verschlüsselung (zum Beispiel Muster und Häufigkeiten in einer Sprache, Leerzeichen, Koinzidenzindex Kappa) – Beschreibung ausgewählter Algorithmen <ul style="list-style-type: none"> • Cäsarchiffrierung • Vigenerechiffrierung • DES (Data Encryption Standard) • Triple-DES • IDEA (International Data Encryption Algorithm) • RSA (nach Ronald Rivest, Adi Shamir, Leonard Adleman) – es ist den Schülern zu verdeutlichen, dass Hilfsmittel zur Verschlüsselung gleichzeitig beim Dechiffrieren eingesetzt werden – Hilfsmittel zur Entschlüsselung (Kasiski-Test, Friedmann-Test) – Freeware- und Sharewareangebote nutzen
<ul style="list-style-type: none"> – Praktische Grundlagen der Kryptographie sowie der Kryptoanalyse und Implementierung ausgewählter Verfahren in der gewählten Programmiersprache 	<ul style="list-style-type: none"> – Erzeugung von Zufallszahlenfolgen – Algorithmen zur Primfaktorenzerlegung – Chiffrier- und Dechiffrieralgorithmen (Cäsar und Vigenere) – Lösen von Zahlenkreuzworträtseln

Wahlthema 7: Datenbankanwendungen zur dynamischen Webseitengenerierung

ZRW: 30 Std.

Vorbemerkungen/Qualifikationen

Im Rahmen dieses Themas werden, ausgehend von einfachen dynamischen Webseiten, Lösungen für den Datenbankzugriff erstellt. Die Implementierung der Abfragen in eine Webseite ist ebenso Schwerpunkt dieses Abschnitts wie die Pflege und Verwaltung von Datenbanken.

Die Schülerinnen und Schüler

- kennen Beispiele für dynamische Webseiten, deren Einsatzgründe, Möglichkeiten und Grenzen,
- kennen die einzelnen Module, die für den Aufbau dynamischer Webseiten notwendig sind,
- können in einer (X)HTML-Seite Formulare einbinden und mit der verwendeten Programmiersprache mit dem Benutzer in Interaktion treten,
- können eine Datenbank erstellen,
- können Abfragen mit einer hierfür geeigneten Sprache (SQL) in dieser Datenbank mit geeigneten Tools vornehmen,
- kennen die Grundlagen des Verbindungsaufbaus zwischen der Datenbank und der Webseite (ODBC, JDBC ...) und können vorhandene Treiber implementieren,
- können die Abfrageergebnisse mit einer hierfür geeigneten Sprache (PHP3/4, ASP, JSP, Java-Servlets, ...) in eine Webseite implementieren.

Inhalte	Hinweise zum Unterricht
<ul style="list-style-type: none"> – Generierung einfacher dynamischer Webseiten (Formulare, Ereignisse) 	<ul style="list-style-type: none"> – Einsatz von JavaScript, VBScript, CSS ... – verschiedene Methoden (post, get), Parameterübergabe
<ul style="list-style-type: none"> – Typische Einsatzbereiche für Datenbanken – Entwurf einer Datenbank 	<ul style="list-style-type: none"> – typische Einsatzbereiche – Datenschutz und Datenmissbrauch – MySQL, Access – die Administration einer Datenbank sollte der Lehrkraft oder dem Systemadministrator obliegen – Beachtung der Grundsätze der Datenbankentwicklung – Festlegen des Datentyps der

	definierten Felder
Inhalte	Hinweise zum Unterricht
	<ul style="list-style-type: none">– Festlegen der Primär- und Fremdschlüssel– Indizierung
<ul style="list-style-type: none">– Abfragen in Datenbanken– Grundlagen des Verbindungsaufbaus	<ul style="list-style-type: none">– Abfragetechniken– Auswahl-, Update- und Löschanfragen– Programmieren der Abfragen und Test mit Hilfe geeigneter Tools– Erläuterung und Beispiel
<ul style="list-style-type: none">– Implementierung der Datenbankabfrage in eine Webseite	<ul style="list-style-type: none">– Erstellung der Webseite, Auswertung der Formularelemente, Erstellung der Abfrage, Darstellung der Ergebnisse in der folgenden (dynamisch erstellten) Seite

Schuljahrgang 13/2**Thema: Projektarbeit****ZRW: 15 Std.****Vorbemerkungen/Qualifikationen**

Das Projekt ist in der Regel aus dem im Halbjahr 13/1 behandelten Wahlthema abzuleiten. Die Projektarbeit soll den im Thema „Informatisches Modellieren“ erlernten Phasen der Softwareentwicklung entsprechen. Wert ist auf folgende Sachverhalte zu legen:

- Förderung der Gruppenarbeit,
- Entwicklung von Sozialkompetenz,
- Anlegen und Fortschreibung der Projektdokumentation,
- formale Beschreibung der Schnittstellen der Softwaremodule.

Der Umfang der Projekte orientiert sich an der zur Verfügung stehenden Zeit und an der Leistungsfähigkeit der Gruppen. Die Ergebnisse der Projektarbeit sind im Rahmen des Unterrichts vorzustellen. Um eine realistische Projektbewertung zu ermöglichen, sind die Projektleiter und Verantwortlichkeiten in der Gruppe festzulegen. Dokumentationen sind als Nachweis für die Eigenleistung von jedem Gruppenmitglied zu führen. Die Festlegung der Projektthemen sowie die Gruppeneinteilung sollte spätestens im zweiten Drittel des Kurshalbjahres 13/1 erfolgen.