

## Einleitung

Im neuen Rahmenlehrplan Informatik für die gymnasiale Oberstufe heißt es (S. 20 f.): „Vertiefungsgebiete sind eine Möglichkeit, exemplarisch klassische und / oder aktuelle Themenbereiche der Informatik zu erarbeiten, um die Einsichten und Kompetenzen der Lernenden zu erweitern. Die Auswahl der Vertiefungsgebiete im Rahmen des schuleigenen Curriculums obliegt den Fachkonferenzen und berücksichtigt das Profil der Schule. [...] Detaillierte Vorschläge für Vertiefungsgebiete finden sich in den Handreichungen zum Rahmenlehrplan Informatik.“

Diese Zusage wird mit dem vorliegenden Papier eingelöst. Auf den folgenden Seiten finden sich – im Unterricht erprobte – Vorschläge zu elf möglichen Vertiefungsgebieten, die von erfahrenen Kolleginnen und Kollegen zusammengestellt wurden. Es handelt sich dabei um „Vorschläge“ im wörtlichen Sinn: Für die Ausgestaltung des Unterrichts in den Vertiefungsgebieten sind die jeweiligen Fachkonferenzen verantwortlich, eine Genehmigung der Fachaufsicht Informatik ist nicht mehr erforderlich. Damit wird auch in unserem Fachbereich die Eigenverantwortlichkeit der Schulen gestärkt.

Auf der anderen Seite kann die Ausarbeitung schuleigener Curricula sehr aufwändig sein – besonders angesichts der vielfachen Belastungen der Unterrichtenden. Daher sollen die vorliegenden Handreichungen auch als Entlastung für die Kolleginnen und Kollegen an den Schulen dienen.

Die bei den einzelnen Gebieten aufgeführten Quellenhinweise (Bücher und Links) können auch im „normalen“ Unterricht zu Rate gezogen werden. Als Beispiel möchte ich auf das Vertiefungsgebiet „Informatik und Gesellschaft“ (I&G) hinweisen, das freundlicherweise von Dr. Jochen Koubek (HU Berlin) zusammengestellt wurde. Bei einem anwendungsorientierten Unterricht wird man auch auf gesellschaftliche und ethische Probleme eingehen wollen. Die bei I&G aufgeführten Quellen sind eine Fundgrube für alle, die nach passenden Hintergrundinformationen suchen.

Es ist beabsichtigt, zu allen Vertiefungsgebieten sowie zu den anderen Schwerpunkten des neuen Rahmenlehrplans Informatik Bereiche auf dem Berliner Bildungsserver einzurichten, damit die Materialien online verfügbar sind und zeitnah aktualisiert werden können. Alle diese Bereiche werden von „Paten“ betreut werden, die dann auch als Ansprechpartner für die jeweiligen Gebiete zur Verfügung stehen. Im Namen aller Paten möchte ich Sie jetzt schon bitten, die Kommunikationsplattform des Berliner Bildungsservers intensiv zu nutzen und Ihre eigenen Erfahrungen einzubringen um die Qualität des Informatikunterrichts in allen Schulen zu fördern.

An der Ausarbeitung dieser Handreichungen waren folgende Kolleginnen und Kollegen aus der ehemaligen Rahmenlehrplankommission Informatik, aus der BICS sowie aus Schulen, die seit einigen Jahren Leistungskurse Informatik anbieten, in unterschiedlicher Weise beteiligt: Werner Arnhold, Alexander Dietz, Klaus Drescher, Jürgen Frank, Walter Gussmann, Dr. Bernd Kokavec, Dr. Eberhard Lehmann, Johann Penon, Ralf Punkenburg, Marielott Schatte-Vogelsang und Christian Steinbrucker. Allen sei auf diesem Weg noch einmal ganz herzlich gedankt. Ich bin sicher, dass Sie sich nach der Lektüre dieser Materialien meinem Dank anschließen werden.

Helmut Witten  
Vorsitzender der  
Rahmenlehrplankommission  
Informatik Sek II  
im Schuljahr 2003/04

## Inhaltsverzeichnis

Einleitung .....	1
A Funktionale Programmierung .....	3
B Logische Programmierung.....	7
C Theoretische Informatik .....	9
D Kryptologie und Datensicherheit.....	13
E Datenbanken .....	15
F Computergrafik.....	19
G Netzwerke .....	23
H Künstliche Intelligenz (KI).....	27
I Technische Informatik .....	29
K Maschinennahe Programmierung.....	31
L Informatik und Gesellschaft.....	35
Impressum.....	40

## A Funktionale Programmierung

Funktionale Programmiersprachen lassen etwas von der Eleganz, Klarheit und Präzision der Mathematik in die Welt der Programmierung einfließen. Im Mittelpunkt steht das mathematische Konzept von Funktionen und Rekursion. Innovative Konzepte, wie Musteranpassung, algebraische Datentypen, Funktionen höherer Ordnung und Polymorphie, machen funktionale Programme elegant und flexibel; sie eignen sich deshalb besonders gut zur Spezifikation.

Die funktionale Programmierung beschäftigt sich nicht mit der Abbildung eines Problems auf den sequentiellen Ablauf eines Programms. Vielmehr steht die Problemlösung (der Algorithmus) im Mittelpunkt (deklarative Programmierung). Dies führt zu einem Programmierstil, der sich grundlegend vom imperativen Programmieren unterscheidet. Die Implementierungen sind in fast allen Fällen kurz und übersichtlich.

Durch die Verwendung einer funktionalen Programmiersprache lassen sich viele zentrale Inhalte des Informatik-Unterrichts an der Schule sehr zielgerichtet thematisieren. Gleichzeitig erfordern diese Sprachen eine bedachte und wohlüberlegte Arbeitsweise, die zur Herausbildung eines sauberen Programmierstils beitragen kann. In Lerngruppen mit heterogenen Vorkenntnissen hat sich die frühzeitige Einführung einer „neuen“ Programmiersprache als förderlich erwiesen.

Im Grundkurs bietet sich das Wahlgebiet bei der Einführung der Rekursion an. Hier kann man an den funktionalen Leitgedanken des Mathematikunterrichts anknüpfen und den Funktionsbegriff zur Beschreibung von Algorithmen erweitern. Die Kenntnis von Rekursion und Listen ist bei der späteren imperativen Behandlung hilfreich.

Im Leistungskurs können funktionale Sprachen frühzeitig eingeführt und dann unterrichtsbegleitend zur Spezifikation eingesetzt werden. Insbesondere für die Wahlgebiete Theoretische Informatik und Kryptologie bieten sich hier interessante Aspekte an, die bei einem späteren Studium nützlich sind (Programmiersprachen, reguläre Ausdrücke, Compiler, Algebraische Spezifikation, Techniken zur Entwicklung korrekter Software usw.).

### Funktionale Programmierung

#### Vorschläge für Schwerpunktvorhaben:

- Grundlegende Sprachelemente funktionaler Sprachen
- Auswertung durch Reduktion
- Wiederholung durch Rekursion
- Listen und Bäume als rekursive Datenstrukturen
- Polymorphie und Funktionskomposition
- Vergleich von funktionalen / imperativen Paradigmen

Fachinhalte	Kompetenzen	Vernetzungen
<p><b>Funktionen als Programme</b></p> <ul style="list-style-type: none"> <li>• Arbeitsumgebung funktionaler Sprachen (Interpreter)</li> <li>• Definition und Anwendung einfacher Funktionen</li> <li>• Typkonzept funktionaler Sprachen Typableitung einer Funktion, Signaturen</li> <li>• manuelle Auswertung durch Reduktion von Funktionstermen</li> </ul>	<ul style="list-style-type: none"> <li>• Vertraut werden mit den Grundlagen des funktionalen Sprachparadigmas</li> <li>• Lösungen für einfache Probleme entwerfen und prüfen</li> <li>• Den Typ einfacher Funktionen ableiten und manuelle Auswertung vornehmen können</li> <li>• Typische Besonderheiten</li> </ul>	<ul style="list-style-type: none"> <li>• Aufgreifen des Funktionsbegriffs aus der <b>Mathematik</b></li> <li>• Funktionen mit mehreren Variablen als Bausteine z.B. zur Formelberechnung (CAS)</li> <li>• Funktionale Zusammenhänge in anderen Unterrichtsfächern</li> <li>• Klare Trennung zwischen</li> </ul>

<ul style="list-style-type: none"> <li>• Funktionen mit mehreren Variablen</li> <li>• Fallunterscheidungen; Musteranpassung</li> <li>• lokale Definitionen (aus Gründen der Effizienz und der besseren Übersicht)</li> </ul>	<p>der funktionalen Programmierung erfassen und beurteilen</p>	<p>(beweisbar korrekten) Berechnungen und Interaktionen, die ggf. Seiteneffekte aufweisen könnten</p>
<p style="text-align: center;"><b>Rekursion</b></p> <ul style="list-style-type: none"> <li>• rekursive Funktionen und deren Auswertung</li> <li>• endrekursive Funktionen und Iteration</li> <li>• Akkumulatoren</li> <li>• Beispiele für rekursive Algorithmen (Türme von Hanoi, Quicksort, Euklid, Linialgorithmus,...)</li> </ul>	<ul style="list-style-type: none"> <li>• Rekursive Algorithmen entwickeln und prüfen</li> <li>• Die Rekursion als Möglichkeit für Wiederholungsanweisungen anwenden können</li> <li>• Rekursive und iterative Problemlösungen untersuchen und bewerten, auch unter dem Gesichtspunkt der Effizienz</li> </ul>	<ul style="list-style-type: none"> <li>• Vertiefung des grundlegenden Begriffes der Rekursion in der Mathematik (Wurzelberechnung, Folgen, Induktion), Chaostheorie</li> <li>• Rekursion und vollständige Induktion (Mathematik)</li> <li>• Rekursionen als wesentlicher Bestandteil einiger Entwurfsstrategien (z.B. Teile und Herrsche)</li> <li>• große Bedeutung in der theoretischen Informatik (Compilerbau)</li> <li>• Vertiefung der Komplexitätstheorie</li> </ul>
<p style="text-align: center;"><b>Rekursive Datenstrukturen</b></p> <ul style="list-style-type: none"> <li>• Die Liste als rekursive, polymorphe Datenstruktur</li> <li>• Arbeiten mit Listenfunktionen (traversieren, suchen, sortieren, einfügen, ...)</li> <li>• Spezifikation von Stack, Queue</li> <li>• Binärbäume als algebraische Datentypen mit Zugriffsfunktionen</li> <li>• mögliche Erweiterungen: Operatorbäume zur Termberechnung, Huffman-Baum zur Datenkomprimierung, Vielwegbäume (Filesystem)</li> </ul>	<ul style="list-style-type: none"> <li>• Typische Datenstrukturen handhaben</li> <li>• Datenstrukturen entwickeln und auswählen</li> <li>• Die Beziehung zwischen Algorithmus und Datenstruktur analysieren</li> <li>• Abstraktion und Modularisierung beurteilen</li> </ul>	<ul style="list-style-type: none"> <li>• Verwendung der Datenstrukturen z.B. im Betriebssystem und in Anwendungsprogrammen</li> <li>• Modularisierung und Abstraktion bei der Softwareentwicklung</li> <li>• Polymorphie in imperativen Programmiersprachen</li> </ul>
<p style="text-align: center;"><b>Funktionen höherer Ordnung</b></p> <ul style="list-style-type: none"> <li>• Funktionen, die Funktionen als Parameter verarbeiten (map, filter, fold, ...)</li> <li>• Funktionskomposition</li> </ul>	<ul style="list-style-type: none"> <li>• Den Funktionsbegriff vertiefen</li> <li>• Modularisierung durch Komposition von Funktionen beherrschen</li> </ul>	<ul style="list-style-type: none"> <li>• Operatoren in der Mathematik, z.B. Differential- und Integraloperatoren</li> <li>• Verkettung von Funktionen in der Mathematik</li> </ul>
<p style="text-align: center;"><b>Vertiefung und Projektarbeit</b></p> <ul style="list-style-type: none"> <li>• Mögliche Vertiefungen: <ul style="list-style-type: none"> <li>- Backtracking</li> <li>- Faltungsfunktionen</li> <li>- Korrektheit von Programmen</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Grundlagen des Softwareentwurfs reflektieren</li> <li>• Korrektheit informatischer Problemlösungen beurteilen</li> </ul>	<ul style="list-style-type: none"> <li>• Funktionale Sprachen als Spezifikationssprache</li> <li>• Korrektheitsüberlegungen beruhen auf dem Prinzip der vollständigen Induktion.</li> </ul>

<ul style="list-style-type: none"> <li>- Interaktion durch Monden</li> <li>• Mögliche Projektthemen:             <ul style="list-style-type: none"> <li>- Rekursionsgrafiken (z.B. mit Turtlegrafik)</li> <li>- Kryptologie</li> <li>- Huffman-Kodierung</li> <li>- Simulation von Automaten</li> </ul> </li> </ul>		<b>(Mathematik)</b>
---	--	---------------------

## Kommentierte Bücher- und Linkliste

### Bücher

- **Bird, Richard, Wadler, Philip.** *Einführung in die funktionale Programmierung.* Hanser, München, Wien, 1992. ISBN 0-13-484197-2  
Eine sprachunabhängige Einführung in die Grundlagen funktionaler Programmierung. Das Buch enthält zahlreiche Beispiele, wobei die deutsche Übersetzung nicht immer gelungen ist. Die Originalausgabe (auf englisch) ist hier besser.
- **Thompson, Simon.** *Haskell - The Craft of Functional Programming.* Addison Wesley, Longman, Harlow England, Reading Mass., second edition 1999. ISBN 0-201-34275-8  
Es handelt sich um eine gut zu lesende Einführung in die Prinzipien der funktionalen Programmierung. Die Beispiele des Buches sind online verfügbar:  
<http://www.cs.ukc.ac.uk/people/staff/sjt/craft2e/>
- **Chakravarty, Manuel, Keller Gabriele.** *Einführung in die Programmierung mit Haskell.* Pearson Education, München, 2004. ISBN 3-8273-7137-6  
Ein preisgünstiges Buch, das alle Aspekte funktionaler Programmierung in knapper Form für den Unterricht abdeckt.
- **Hudak, Paul.** *The Haskell School of Expressions.* Cambridge University Press 2000.  
Das Buch erklärt die Konzepte funktionaler Programmierung an nicht trivialen Beispielen aus dem Multimedia-Bereich. Die Beispiele sind für Anfänger weniger gut geeignet. Unterlagen findet man unter <http://www.haskell.org/soe>.
- **Pepper, Peter.** *Funktionale Programmierung in OPAL, ML, HASKELL und GOFER.* Springer-Verlag, Berlin, Heidelberg, New York, 2000.  
Eines der wenigen deutschen Bücher, das mehrere Sprachen vorstellt.
- **Rabhi, Fethi; Lapalme, Gui.** *Algorithms: A functional programming approach,* Addison-Wesley, 1999. ISBN 0-201-59604-0  
Der Autor beschreibt die traditionellen Algorithmen mit Hilfe des funktionalen Paradigma. Als Sprache wird Haskell verwendet. Das Buch ist trotz der englischen Sprache leicht verständlich.
- **Goos, Gerhard.** Vorlesungen über Informatik Band 1: Grundlagen und funktionales Programmieren. Springer-Verlag, Berlin, Heidelberg, New York, 2000. ISBN 3-540-67270-2.  
„Dieses Lehrbuch ist eine moderne Einführung in die Informatik für das Grundstudium an Universitäten und Fachhochschulen. Der Autor stellt die Prinzipien der „Systemmodellierung“ und der „ingenieurmäßigen Softwarekonstruktion“ in den Mittelpunkt seiner Darstellung. Der erste Band bietet den Einstieg in den Vorlesungszyklus und vermittelt die Grundbegriffe, mit denen die Informatik umgeht, und die theoretischen Grundlagen (Algebren, Relationen, Logikgrundlagen, funktionales Programmieren, abstrakte Datentypen), die benötigt werden, um das systematische Konstruieren von Informatiksystemen zu erlernen.“

### Links

- **<http://www.haskell.org>**  
Diese Seite ist der Startpunkt für alle Themen rund um die Sprache HASKELL. Dort sind auch Links zum Interpreter HUGS sowie eine umfangreiche Literaturliste zu finden.
- **<http://www.haskell.org/tutorial>**  
Paul Hudak, John Peterson, and Joseph H. Fasel.

Der Artikel gibt einen Überblick über das Typsystem und die Spezialitäten von Haskell. Dabei wird teilweise Vorwissen vorausgesetzt.

- **<http://www-106.ibm.com/developerworks/edu/os-dw-linuxhask-i.html>**  
from IBM developerWorks  
Dieser Artikel beschreibt das Paradigma funktionaler Programmierung und richtet sich an Programmierer, die bereits eine iterative Sprache kennen. Als Sprache wird Haskell verwendet. (Der Artikel setzt eine kostenlose Anmeldung voraus.)
- **<http://www.isi.edu/~hdaume/htut/>**  
Eine umfangreiche Darstellung der Sprachkonzepte auf englisch. (PDF / PS -Datei).
- **[http://www.informatik.uni-bonn.de/~ralf/teaching/Hskurs\\_toc.html](http://www.informatik.uni-bonn.de/~ralf/teaching/Hskurs_toc.html)**  
Ein Online-Kurs von Dr. Ralf Hinze (Uni Bonn) der viele Themen und Aspekte abdeckt. Einige Teile sind für Anfänger weniger geeignet.
- **[http://www.cs.mu.oz.au/tr\\_submit/test/cover\\_db/mu\\_TR\\_1999\\_14.html](http://www.cs.mu.oz.au/tr_submit/test/cover_db/mu_TR_1999_14.html)**  
Haskell for Miranda Programmers  
Der Autor beschreibt sehr umfassend die Unterschiede und Stolperfallen beim Umstieg von Miranda auf Haskell (28 Seiten auf englisch).
- **<http://bscw.gmd.de/pub/bscw.cgi/0/44857597>**  
Herr Dr. Puhlmann stellt hier eine ausführliche Einführung in das funktionale Programmieren mit Haskell zur Verfügung (PDF-Datei).
- **<http://www.pns-berlin.de/haskell/>**  
Auf dieser Seite werden Unterrichtsprojekte des Profil-, Grund und Leistungskurses vorgestellt. Zusätzlich finden sich hier Unterlagen für mehrere Haskell-Fortbildungen für Lehrer.
- **<http://www.pns-berlin.de/material/haskelbroschuere.pdf>**  
Diese Broschüre gibt einen kurzen Überblick über die wichtigsten Eigenschaften von Haskell. Zu jedem Kapitel sind einige Übungsaufgaben enthalten. Für Schüler geeignet.
- **<http://www.cs.kent.ac.uk/pubs/2000/958/>**  
Regular Expressions and Automata using Haskell  
Der Artikel beschreibt, wie reguläre Ausdrücke mit endlichen nichtdeterministischen Automaten unter Haskell implementiert werden. Zugleich wird erläutert, wie diese Automaten in deterministische umgewandelt werden. (29 S., englisch)
- **<http://www.mathematik.uni-marburg.de/~priebe/lehre/ws9900/fp/inf3bskript99klein.pdf>**  
Skript zu einer zweistündigen Vorlesung (Uni-Marburg).
- **<http://www.cs.kent.ac.uk/people/staff/sjt/craft2e/errors.html>**  
Übersicht über die typischen Fehlermeldungen in Haskell.
- **<http://paris.cs.uni-magdeburg.de/lehre/ws-03-04/pkm-mat/slides/index.html>**  
Folien zur Vorlesung: Programmierkonzepte und Modellierung von Prof. Dr. D. Rösner (WS 2003/04). Die Folien zeigen einen guten Überblick über die unterschiedlichen Sprachkonzepte funktionaler, logischer und imperativer Sprachen.

## B Logische Programmierung

Wie die Funktionale Programmierung beschäftigt sich auch die Logische Programmierung nicht mit der Abbildung eines Problems auf den sequentiellen Ablauf eines Programms. Vielmehr steht die Problemlösung im Mittelpunkt einer deklarativen Modellierung. Dies impliziert ein Programmier - Paradigma, das sich grundlegend vom imperativen Programmieren unterscheidet. Die Behandlung als Vertiefungsgebiet eröffnet in geeigneter Weise die Möglichkeit, dieses Paradigma im anschließenden Softwareprojekt fortzuführen und anzuwenden.

Mit PROLOG steht eine bewährte Programmiersprache zur Verfügung (siehe Anhang).

Logische Programmierung		
Vorschläge für Schwerpunktvorhaben: <ul style="list-style-type: none"> <li>• Wissensbasis aus Beschreibungen von Fakten und Regeln</li> <li>• Nachweis der Un/erfüllbarkeit einer Menge von Klauseln</li> <li>• „Lernfähigkeit“ eines Expertensystems</li> <li>•</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<b>Wissensbasis</b> <ul style="list-style-type: none"> <li>• Fakten</li> <li>• Regeln</li> <li>• Anfragen</li> <li>• rekursive Prädikate</li> </ul>	<ul style="list-style-type: none"> <li>• Beschreibung von Objekten und ihren Eigenschaften in der Wissensbasis</li> <li>• Konsultation der Wissensbasis</li> <li>• Anfragen entwickeln und durchführen</li> </ul>	<ul style="list-style-type: none"> <li>• Der Unterschied bei der Implementation mit logischer und imperativer Programmiersprache wird deutlich.</li> </ul>
<b>Strategien der Problemlösung</b> <ul style="list-style-type: none"> <li>• Unifikation, Resolution, Backtracking, Cut</li> <li>• Suchverfahren (Graphen, Breitensuche, Tiefensuche und Heuristische Suche)</li> </ul>	<ul style="list-style-type: none"> <li>• Die wesentlichen Verfahren der PROLOG - Beweisführung benennen, einsetzen und vergleichen</li> <li>• Selbstmodifizierende Klauseln zur Problemlösung einsetzen</li> </ul>	<ul style="list-style-type: none"> <li>• Effiziente und selbst lernende Programme schreiben zu können, eröffnet den Einblick in die Thematik der <b>Künstlichen Intelligenz</b> (Weiteres Vertiefungsgebiet).</li> </ul>
<b>Listen</b> <ul style="list-style-type: none"> <li>• Terme</li> <li>• rekursive Struktur von Listen</li> <li>• Kopf und Rest</li> </ul>	<ul style="list-style-type: none"> <li>• Struktur von Listen erkennen und in Klauseln einsetzen</li> <li>• Datenrepräsentationen zwischen Argument- und Listenform entscheiden und implementieren</li> </ul>	<ul style="list-style-type: none"> <li>• Die Unterschiede der Listenimplementationen zwischen logischen und imperativen Programmiersprachen werden deutlich.</li> </ul>
<b>Anwendung der Logischen Programmierung als Entwurfswerkzeug</b> <ul style="list-style-type: none"> <li>• Erste Prototypen für Anwendungsprogramme aus den folgenden informatischen Kernthemen, z. B.:</li> <li>• Automaten</li> <li>• Turingmaschinen</li> </ul>	<ul style="list-style-type: none"> <li>• Klärung des Automatenbegriffs und Modellierung in Prolog, insb. am Beispiel des Kellerautomaten</li> <li>• Untersuchen einer Modellierung der Turing-Maschine</li> </ul>	<ul style="list-style-type: none"> <li>• Computer-Linguistik: Einsicht in die Natur sprachlicher Kommunikation und in die Struktur der Sprache gewinnen (<b>Deutsch und Fremdsprachen</b>)</li> </ul>

<ul style="list-style-type: none"><li>• Parser und Interpreter</li><li>• maschinelle Sprachverarbeitung</li><li>•</li></ul>	<ul style="list-style-type: none"><li>• Analysieren der Syntax einer Sprachbeschreibung durch ein Parser-Programm</li><li>• maschinelle Verarbeitung natürlicher Sprache überprüfen</li></ul>	
---	---	--

### **Kommentierte Bücher- und Linkliste**

#### ***Schulbuch***

- **Gerhard Röhner:** *Informatik mit Prolog*. Hessisches Landesinstitut für Pädagogik, ISBN 3-88327-499-2, zweite vollständig neu überarbeitete Auflage 2002  
Dieses Buch für 12.- €deckt alle beschriebenen Themen und Inhalte voll ab.

#### ***Fachbücher***

- **Cloksin/Mellish:** *Programmieren in Prolog*. 4.Auflage. Springer-Verlag, Heidelberg, 1997
- **Peter Schnupp:** *PROLOG – Einführung in die Programmierpraxis*. Carl Hanser Verlag, 1986
- **Bothe / Stojanow:** *Praktische Prolog-Programmierung*. Verlag Technik Berlin, 1990
- **Cordes/Kruse/Langendörfer/Rust:** *Prolog*. (3. Auflage), Vieweg Verlagsgesellschaft, 1992

#### ***Linkliste***

- **<http://www.bildung.hessen.de/abereich/inform/skii/material/prolog.htm>**  
homepage zum Schulbuch, s.o.
- **<http://www.informatica-didactica.de/Lehre/FunktionalLogisch>**  
Uni-Skripte mit Aufgabe und Lösungen
- **<http://www.learn-line.nrw.de/angebote/werkstattbw/prolog/docs/aufgaben>**  
Arbeitsmaterialien (Arbeitsblätter, Projektaufgaben, Klausuraufgaben)

## C Theoretische Informatik

Theoretische Informatik gehört zu den Kernthemen der Schulinformatik. Sie legt ein zeitbeständiges theoretisches Fundament zum Informatikunterricht und hat eine allgemeinbildende Wirkung (z. B. Erfassung prinzipieller Grenzen der Informationstechnik). Bei ihrer Behandlung in der Schule sollen die theoretischen Probleme aus Anwendungsproblemen heraus entstehen. Eine davon abgehobene Betrachtungsweise ist nicht intendiert. Damit wird z. B. ein Einstieg in die Problematik mit Hilfe von Automatenmodellen nahe gelegt. Ein anderer motivierender Einstieg kann über das Busy-Beaver-Problem und damit über Turing-Maschinen erfolgen. Angesichts der theoretischen Aspekte ergeben sich Querverbindungen zur Mathematik, die besonders im Leistungskurs Informatik oder Mathematik (z.B. Markow-Ketten) genutzt werden können.

Es werden drei Themen vorgeschlagen, die sich leicht vernetzen lassen.

Theoretische Informatik		
Vorschläge für Schwerpunktvorhaben:		
<ul style="list-style-type: none"> <li>• Beschreibung von Systemen durch endliche Automaten</li> <li>• Turing-Maschinen</li> <li>• Typen und Grammatiken formaler Sprachen</li> <li>•</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<p style="text-align: center;"><b>Endliche Automaten</b></p> <ul style="list-style-type: none"> <li>• Simulation von realen Automaten durch endliche Automaten (Transduktoren)</li> <li>• erkennende Automaten (Akzeptoren)</li> <li>• Zustandsgraphen, Zustandstabellen,</li> <li>• Simulation von Automaten durch eine geeignete Programmiersprache</li> </ul>	<ul style="list-style-type: none"> <li>• Mit formalen Beschreibungsmethoden (Automat, Grammatik) umgehen können</li> <li>• Überblick über Einsatzbereiche realer Automaten gewinnen</li> <li>• Verstehen der Arbeitsweise von Automaten aus dem Alltag</li> <li>• Modellieren realer Automaten durch geeignete Darstellungsformen (Tabellen, Zustandsgraphen, Programme...)</li> <li>• Erkennen, welche Probleme mit Automaten lösbar sind und welche nicht.</li> </ul>	<ul style="list-style-type: none"> <li>• Automaten spielen im Alltag eine wichtige Rolle</li> <li>• Automatentheorie ist wichtig u.a. für Texterkennung und im Compilerbau</li> <li>• Zustandsgraphen werden in vielen Gebieten für die Beschreibung von Zustandsänderungen und die Klassifizierung von Zustandsmengen verwendet</li> <li>• Unterschied zwischen Determinismus und nicht-deterministischen Methoden</li> <li>• Weitere Vernetzungsmöglichkeiten: formale Sprachen Turing-Maschine Registermaschine andere Rechenmodelle</li> </ul>
<p style="text-align: center;"><b>Formale Sprachen und Grammatiken</b></p> <ul style="list-style-type: none"> <li>• Syntax und Grammatiken natürlicher und formaler Sprachen Begriffe: Terminalsymbol, Nichtterminalsymbol, Startsymbol, Produktionsregel, Wort, Satz</li> <li>• Synthese und Analyse</li> </ul>	<ul style="list-style-type: none"> <li>• Unterscheidung zwischen Syntax und Semantik</li> <li>• Beschreibung einfacher Grammatiken in geeigneter Form</li> <li>• Konstruieren formaler Sprachen</li> <li>• Verstehen der</li> </ul>	<ul style="list-style-type: none"> <li>• Ggf. Anknüpfung an das Thema "Endliche Automaten"</li> <li>• Anknüpfung an den Sprachunterricht (Linguistik)</li> <li>• Anwendung von Sprachen in unterschiedlichem</li> </ul>

<ul style="list-style-type: none"> <li>• Syntaxdiagramme</li> <li>• Modellierung einfacher Sprachen</li> <li>• Parser für formale Sprachen</li> <li>• kontextfreie Sprachen</li> <li>• Backus-Naur-Formalismus (BNF)</li> <li>• Zusammenhang zwischen regulären Sprachen und endlichen Automaten, Kellerautomat</li> <li>• Compiler, Interpreter</li> </ul>	<p>Funktionsweise eines Parsers</p> <ul style="list-style-type: none"> <li>• Gewinnen von Überblick über die grundsätzliche Arbeitsweise von Compilern und Interpretern</li> </ul>	<p>Umfeld: Gerätekommunikation (Protokolle), Maschinensprache, Programmiersprachen, mathematische Formelsprache, natürliche Sprachen</p> <ul style="list-style-type: none"> <li>• philosophische Aspekte: Roboter / Mensch; Maschine / Mensch</li> </ul>
<p style="text-align: center;"><b>Turingmaschinen</b></p> <ul style="list-style-type: none"> <li>• Architektur und Arbeitsweise einer Turingmaschine</li> <li>• Turing-Programme (Algorithmen)</li> <li>• Beispiele wie <math>a:=a+1</math>; Multiplikation <math>a*b</math> (<math>a,b</math> aus <math>\mathbb{N}</math>), Kopieren eines Wortes</li> <li>• Busy Beaver</li> <li>• Turing-Maschine mit Darstellungsformen: Turing-Befehle z.B. als 5-er Tupel (aktueller Zustand, gelesenes Zeichen, Richtung der Bandbewegung, neu erreichter Zustand), Zustandsgraphen</li> <li>• Vergleich: Turing-Maschine mit endlichem Automat/mit Computer</li> <li>• Präzisierung des Algorithmenbegriffs, Berechenbarkeit</li> <li>• Halteproblem</li> </ul>	<ul style="list-style-type: none"> <li>• Verstehen der Grundlagen von Informatiksystemen</li> <li>• Spezielle Informatiksysteme (Modell-Turing-Maschine) anwenden können</li> <li>•</li> <li>• Gestaltung von Informatiksystemen</li> <li>•</li> <li>• Prinzipielle Grenzen von Informatiksystemen erfahren</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>• Im WWW finden sich Demonstrationsprogramme. Es ist auch möglich, selbst im Unterricht Modelle zu erstellen, die beispielsweise in anderen Kursen nutzbar sind.</li> <li>• Das Busy-Beaver-Problem hat im WWW große Aufmerksamkeit gefunden. Es führt schnell auf Fragen der Berechenbarkeit und auf das Halteproblem.</li> <li>• Zustandsgraphen werden in vielen Bereichen zur Beschreibung von Zustandsveränderungen benutzt, u.a. in der <b>Mathematik</b> bei Markow-Ketten</li> <li>• Vernetzung von Turing-Maschine mit Automaten und Computern</li> <li>• Aufgreifen des Funktionsbegriffs aus der <b>Informatik/Mathematik</b></li> </ul>
<p style="text-align: center;"><b>Mögliche Vertiefungen</b></p>		<ul style="list-style-type: none"> <li>• Einige Themen können auch in vertiefter Form als Projektarbeit durchgeführt werden..</li> </ul>

## Kommentierte Bücher- und Linkliste

### Bücher

- **Burkert / Griesel / Postel:** *Informatik heute*. Band 2, Algorithmen und Datenstrukturen, Schroedel-Schöningh 1988, ISBN 3-507-83091-4  
Älteres Schulbuch u.a. mit den Gebieten aus der Theoretische Informatik (Endliche Automaten, Formale Sprachen, Algorithmen und Berechenbarkeit, S. 82-116), viele Aufgaben, Schülerbuch
- **Lehmann, Eberhard:** Konzeptionelle Überlegungen zur Einbeziehung informatischer Inhalte und Methoden beim Computereinsatz im Mathematikunterricht der Sekundarstufe 2.  
Verlag Franzbecker 2003  
Kapitel 3.3 (S. 100-141): Zustandsgraphen in Informatik und Mathematik (Endliche Automaten und Markow-Ketten, das Busy-Beaver-Problem, Turingmaschinen, Versandproblem, Crapspiel)
- **Brecht, Werner:** *Theoretische Informatik*. Vieweg-Verlag, Braunschweig 1995  
Grundlagen und praktische Anwendungen (u.a. Turing-Maschinen, Berechen- und Entscheidbarkeit, Komplexität, Nebenläufigkeit)
- **Dewdney, A.K.:** *Der Turing-Omnibus - eine Reise durch die Informatik mit 66 Stationen*. Springer-Verlag, Berlin-Heidelberg, 1995 (Zahlreiche "Reiseerlebnisse" zur Technischen Informatik)
- **Gasper, Leiß, Spengler, Stimm:** *Technische und theoretische Informatik*. Bayerischer Schulbuch-Verlag, München 1992 (u.a. Endliche Automaten, Formale Sprachen, Turingmaschinen, Busy-Beaver, Unentscheidbarkeit, Halteproblem, Komplexität)
- **Lehmann, Eberhard:** *Die Turingmaschine im Anfangsunterricht* – ein Bericht von den ersten Stunden eines Informatikkurses in Klasse 11, in LOGIN 1999, Heft 6, S.44-52
- **Werner u.a.:** Taschenbuch der Informatik, Fachbuchverlag Leipzig 1995
- **Lehmann, Eberhard:** *Wahrscheinlichkeitsrechnung*. Problemorientierte Unterrichtseinheiten. Verlag Volk und Wissen, Berlin 1997  
(u.a. Beispiele für Markow-Ketten und Zustandsgraphen)
- **Baumann, Rüdeger:** *Didaktik der Informatik*. 2.Auflage, Klett-Verlag 1996  
(u.a. in Kapitel 21 Abriss über Theoretische Informatik aus didaktischer Sicht)
- **Engelmann, Lutz:** *Informatik bis zum Abitur*. Paetec, Berlin, 2002 (ISBN 3-89818-600-8)  
Das Unterrichtsbuch enthält eine ausführliche Darstellung der theoretischen Informatik mit einigen Beispielaufgaben. Neben formalen Sprachen und Automaten werden die Berechenbarkeitstheorie behandelt und effiziente Algorithmen mit Komplexitätsbetrachtungen vorgestellt.
- **Baumann, Rüdeger:** *Informatik mit Pascal*. Klett-Verlag, Stuttgart 1981 (ISBN 3-12-717740-2)  
Trotz der antiquierten Programmiersprache bietet dieses Buch eine grundlegende Einführung in die Automatentheorie, formale Sprachen, Turingmaschinen. Anhand von Beispielen, die in Pascal implementiert werden, werden die grundlegenden Aspekte klar herausgearbeitet.
- **Schöning, Uwe:** *Theoretische Informatik – kurzgefasst*. Spektrum, Heidelberg, Berlin, 2001. 198 Seiten. (ISBN 3-8274-1099-1)  
Es handelt sich ein überarbeitetes Vorlesungsskript des Autors an der Universität Ulm und reflektiert die Aspekte aus eher theoretischer Sicht. Für Vertiefungen und besondere Aspekte des Themas geeignet. Für Anfänger weniger geeignet.
- **Röhner, Gerhard:** *Informatik mit Prolog*. Hessisches Landesinstitut für Pädagogik, ISBN 3-88327-499-2, zweite vollständig neu überarbeitete Auflage 2002  
In den letzten Kapiteln wird die theoretische Informatik ausführlich behandelt.

### **Links**

- **<http://oszhdl.be.schule.de/gymnasium/faecher/informatik/automaten>**  
Es werden endliche Automaten an Beispielen vorgestellt und die Begriffe erläutert. Das Automatenmodell der ETH-Zürich (KaraToJava) wird u.a. an Beispiel vorgestellt.
- **<http://www.educeth.ch/informatik/karatojava/>**  
Mit dem Kara-Modell können Automaten direkt simuliert werden. Auch Turingmaschinen können damit veranschaulicht werden. Zusätzlich enthalten diese Seiten viele Unterrichtsmaterialien zum Thema.
- **<http://www.tinohempel.de/info/info/ti/informatik.htm>**  
Tino Hempel beschreibt seine Unterrichtserfahrungen im Grund- und Leistungskurs zum Thema. Die Seiten enthalten gut dokumentierte Beispiele, weitere Anregungen und Hinweise. Beispiele sind teilweise in Prolog implementiert.
- **[http://atbruegge27.informatik.tu-muenchen.de/teaching/ws01/GrProg/fohlen/13\\_Automaten.pdf](http://atbruegge27.informatik.tu-muenchen.de/teaching/ws01/GrProg/fohlen/13_Automaten.pdf)**  
Ein knappe, übersichtliche Darstellung der Automatentheorie (PDF-Datei).
- **<http://www.oberstufeninformatik.de/theorie>**  
Startseite von H. Gierhardt zur theoretischen Informatik. Zahlreiche Verweise zu weiterführenden Seiten und Software.
- **[http://rw4.cs.uni-sb.de/~ganimal/GANIFA/page0\\_g.htm](http://rw4.cs.uni-sb.de/~ganimal/GANIFA/page0_g.htm)**  
Ein Online-Lehrbuch zur Generierung endlicher Automaten (Uni Saarbrücken).
- **<http://www.bildung.hessen.de/abereich/inform/skii/lk/theorie.pdf>**  
Ein Autorenteam aus Hessen hat für den hessischen Rahmenplan eine mögliche Behandlung des Themas im Unterricht zusammengestellt. Neben einer Stundenplanung werden einige Beispiele ausführlich vorgestellt. Diese Darstellung bietet in aller Kürze einen guten Überblick über viele Aspekte der theoretischen Informatik.
- **<http://www.pns-berlin.de/lk/ti/index.html>**  
Erkennende Automaten und formale Sprachen werden hier unter dem Aspekt einer Implementierung in Haskell und/oder Java dargestellt. Die Informationen beruhen auf Unterrichtserfahrungen aus dem LK Informatik an der Paul-Natorp-Oberschule.

## D Kryptologie und Datensicherheit

Die Kryptologie bietet vielfältige Themen und Möglichkeiten der unterrichtlichen Behandlung. Der hier vorgestellte Vorschlag, den RSA-Algorithmus in den Mittelpunkt zu stellen ist unterrichtlich erprobt und bietet zahlreiche Querverbindungen und Ausbaumöglichkeiten. Andere Schwerpunkte mit alternativen (auch symmetrischen) Verschlüsselungsverfahren sind sinnvoll und möglich.

Als mathematisches Handwerkszeug wird das modulare Rechnen mit großen ganzen Zahlen benötigt. Einige „klassische“ Algorithmen (Euklid bzw. dessen Erweiterung von Berlekamp, Sieb des Erathostenes etc.) werden nicht als isolierte mathematische Algorithmen, sondern eingebunden in einen Anwendungszusammenhang erarbeitet. Die praktische Bedeutung kryptologischer Algorithmen für die Sicherheit in Netzen muss nicht besonders hervorgehoben werden, aktuelle politische Bezüge ergeben sich z. B. durch die Kryptodebatte („Soll starke Kryptologie für jedermann frei sein?“).

Mit dem skizzierte Unterrichtsgang wird eine genetische Entwicklung vorgeschlagen. Für die Umsetzung sollten Werkzeuge mit Langzahlarithmetik zur Verfügung stehen (z. B. Python oder ein CAS-System wie Derive oder die Klasse „BigInteger“ in Java oder entsprechende Bibliotheken in anderen Programmiersprachen). Auch der Einsatz des freien Programms „Cryptool“ ist empfehlenswert.

Kryptologie und Datensicherheit		
<b>Vorschläge für Schwerpunktvorhaben:</b> <ul style="list-style-type: none"> <li>• Anwendungen der modernen asymmetrischen Kryptographie im Internet</li> <li>• Algorithmen der elementaren Zahlentheorie im Anwendungszusammenhang</li> <li>• Wie funktioniert das RSA-Verfahren? Was ist das RSA-Challenge?</li> <li>• Kryptologie im Spannungsfeld von Freiheit und Sicherheit</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<b>Das Caesar-Verfahren</b> <ul style="list-style-type: none"> <li>• Grundbegriffe der Kryptologie (Ver- und Entschlüsselung, Geheimentextalphabet, Klartextalphabet, ...)</li> <li>• Kryptoanalyse des Caesar-Verfahrens</li> </ul>	<ul style="list-style-type: none"> <li>• Das Caesar-Verfahren als typische monoalphabetische Verschlüsselung kennen und anwenden</li> <li>• Den Caesar-Code durch Ausschöpfen des Schlüsselraums brechen</li> </ul>	<ul style="list-style-type: none"> <li>• Modulare Addition und Subtraktion (<b>Mathematik</b>)</li> <li>• Geschichte der Kryptologie</li> </ul>
<b>Multiplikative Verschlüsselung</b> <ul style="list-style-type: none"> <li>• Ver- und Entschlüsselung durch Modulare Multiplikation</li> <li>• Anwendung des Verschlüsselungsverfahrens auf große Zahlen (Blockchiffre)</li> <li>• Einführung des Prinzips der asymmetrischen Verschlüsselung (Problem der Schlüsselverteilung und der authentischen Kommunikation)</li> <li>• Erweiterter euklidischer Algorithmus zur Bestimmung der modularen Inversen</li> </ul>	<ul style="list-style-type: none"> <li>• Die multiplikative Verschlüsselung (auch mit großen Zahlen) durchführen können</li> <li>• Mit dem Verfahren der Blockchiffrierung vertraut werden</li> <li>• Die Grundprinzipien der asymmetrischen Kryptographie verstehen</li> <li>• Mangelnde Eignung der multiplikativen Verschlüsselung für die asymmetrische Kryptographie abschätzen</li> </ul>	<ul style="list-style-type: none"> <li>• Elementare Zahlentheorie (<b>Mathematik</b>)</li> <li>• Geschichte der Kryptologie: Paradigmenwechsel durch den Übergang zur asymmetrischen Kryptographie</li> </ul>

<p style="text-align: center;"><b>Das RSA-Verfahren</b></p> <ul style="list-style-type: none"> <li>• Chiffrierung durch modulares Potenzieren</li> <li>• Schlüsselgenerierung beim RSA-Verfahren</li> <li>• Nachweis der Korrektheit des RSA-Algorithmus</li> </ul>	<ul style="list-style-type: none"> <li>• Ver- und Entschlüsselung nach dem RSA-Verfahren durchführen</li> <li>• Den „Square and Multiply“ Algorithmus verstehen und anwenden</li> <li>• Das Prinzip zur Bestimmung von (großen) Primzahlen verstehen</li> <li>• Die Einweg-Eigenschaft der Multiplikation großer Primzahlen erfassen</li> <li>• Die Generierung von Schlüsselpaaren durchführen</li> <li>• Die Sicherheit des RSA-Verfahrens beurteilen</li> <li>• Die Korrektheit des RSA-Verfahrens nachvollziehen</li> </ul>	<ul style="list-style-type: none"> <li>• Mögliche Herleitung des Algorithmus über die sog. „ägyptische“ Multiplikation (Geschichte des Rechnens)</li> <li>• Sieb des Erathostenes</li> <li>• Kleiner Satz von Fermat (elementare Zahlentheorie: <b>Mathematik</b>)</li> <li>• Mögliche Erweiterungen: Probabilistisches Verfahren zur schnellen Bestimmung großer Zahlen, die (wahrscheinlich) prim sind Nicht-elementare Methoden zur Primfaktorzerlegung</li> </ul>
<p style="text-align: center;"><b>Anwendungen des RSA-Verfahrens</b></p> <ul style="list-style-type: none"> <li>• Anwendungen und Probleme des RSA-Verfahrens</li> <li>• Zentrale Zertifizierung vs. „Web of Trust“</li> <li>• Schutz der Privatsphäre vs. Schutz vor Kriminellen und Terroristen</li> </ul>	<ul style="list-style-type: none"> <li>• Das Prinzip der digitalen Unterschrift verstehen</li> <li>• Ein Verschlüsselungssystem, das RSA verwendet (PGP o. ä.) nachvollziehen und anwenden</li> <li>• Ein Problembewusstsein für die Schlüsselzertifizierung entwickeln</li> <li>• Interessenkonflikte bei der starken Verschlüsselung abwägen</li> </ul>	<ul style="list-style-type: none"> <li>• Anwendungen z. B. bei Vertragsabschlüssen im Internet (<b>Recht, Wirtschaft</b>)</li> <li>• Die Geschichte von PGP (<b>Politik</b>)</li> <li>• Krypto-Debatte (<b>Politik</b>)</li> <li>• Mögliche Erweiterung: Digitales Geld (<b>Wirtschaft</b>)</li> </ul>

### Kommentierte Bücher- und Linkliste

Diese Listen sowie Materialien aus Fortbildungen von H. Witten finden sich z. Zt. unter <http://bscw.schule.de/pub/bscw.cgi/0/159132>, später werden sie auch auf dem Berliner Bildungsserver zu finden sein.

## E Datenbanken

Während im ersten Unterrichtsjahr der pragmatische Umgang mit einer Datenbank im Vordergrund steht, werden hier die zugrunde liegenden theoretischen Konzepte vertieft behandelt. Die Datenmodellierung ist unabhängig von dem verwendeten Datenbankkonzept. Die weitere Unterrichtsreihe konzentriert sich aber bewusst auf relationale Datenbanken und deren Schnittstelle zum Internet.

Datenbanken		
<b>Vorschläge für Schwerpunktvorhaben:</b> <ul style="list-style-type: none"> <li>• Analyse, Modellierung und Implementation einer Datenbank auf der Grundlage realen eines Anwendungsfalles, z.B. Schlüssel-, Geräte und Raumverwaltung der Schule</li> <li>• Analyse, Wartung und Implementation einer bestehenden Datenbank</li> <li>• Umsetzung eines bisher mit einer Programmiersprachen gelösten Dateiverwaltung mit einer Datenbank</li> <li>•</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<b>Grundlagen von Datenbanksystemen</b> <ul style="list-style-type: none"> <li>• Anwendungs- und Datenunabhängigkeit</li> <li>• Mehrbenutzerbetrieb</li> <li>• 3-Ebenen-Architektur (ANSI/SPARC-Schemaebenen)</li> <li>• Datenbanksystem-Generationen (Netzwerk-, hierarchische Systeme)</li> <li>• Relationale Datenmodelle</li> <li>• Objektrelationale Datenmodelle</li> </ul>	<ul style="list-style-type: none"> <li>• Schichtenarchitektur als zentrales Prinzip der Informatik erkennen</li> <li>• Trennung von Information und Repräsentation als tragendes Prinzip der Informatik erkennen und beschreiben</li> </ul>	<ul style="list-style-type: none"> <li>• Externe, konzeptuelle und interne Ebene von DBS</li> <li>• Geschichte der Informatik</li> </ul>
<b>Modellierung mit dem Entity-Relationship-Modell</b> <ul style="list-style-type: none"> <li>• vertiefte Behandlung der schematischen Darstellung von Datenbankstrukturen z.B. ERM oder UML</li> <li>• Klassifizierung der Beziehungsarten (eindeutig (1:1), eindeutig (1:n), komplex (n:m), zwingend oder freigestellt</li> <li>• Primär- und Fremdschlüssel</li> </ul>	<ul style="list-style-type: none"> <li>• Ein Datenverwaltungsproblem der realen Welt in ein schematisches Modell überführen, z.B. in ein ERM oder ein UML-Diagramm</li> <li>• Die Abhängigkeit des ER-Modells von realen Problemlagen benennen und situationsgerecht in ein entsprechendes ERM umsetzen</li> <li>• Das ER-Modell mit Klassendiagrammen vergleichen</li> </ul>	<ul style="list-style-type: none"> <li>• Umsetzung von Problemstellungen aus anderen Fächern</li> <li>• Nutzung von Software zum kooperativen Arbeiten (Groupware)</li> <li>• Visualisierungsprogramme zum Erstellen des schematischen Datenmodells benutzen.</li> </ul>

<p style="text-align: center;"><b>Umsetzung des Entity-Relationship-Modells in das relationale Datenmodell</b></p> <ul style="list-style-type: none"> <li>• Relationales Datenmodell (Kartesisches Produkt, Teilmenge, n-Tupel, Tabelle, Spalte, Zeile)</li> <li>• Umsetzung von Entitäten und Beziehungen in Relationen (Tabellen) nach Abbildungsregeln</li> <li>• Normalisierung (Beseitigung von Mehrdeutigkeiten und Inkonsistenzen), funktionale Abhängigkeit, 1. - 3. Normalform</li> <li>• Integritätsbedingungen</li> </ul>	<ul style="list-style-type: none"> <li>• Mit Hilfe der Abbildungsregeln aus schematischen Modellen Tabellen erstellen.</li> <li>• Die Normalisierung zur Überprüfung dieser Tabellen benutzen.</li> <li>• Integritätsbedingungen für die Tabellen festlegen und begründen.</li> </ul>	<ul style="list-style-type: none"> <li>• Beziehungen zur Relationentheorie der <b>Mathematik</b></li> </ul>
<p style="text-align: center;"><b>Implementierung des Datenmodells mit der Standard-Datenbanksprache SQL</b></p> <ul style="list-style-type: none"> <li>• Datendefinitionsbefehle und Datenmanipulationsbefehle von SQL</li> <li>• select-Befehl für komplexere Abfragen</li> <li>• Operationen zum Einfügen, Ändern, Löschen und Wiedergewinnen von Daten als Manipulation von Relationen (Vereinigung, Durchschnitt, Teilmenge, Projektion, Verbund, ...)</li> </ul>	<ul style="list-style-type: none"> <li>• Tabellen mit Hilfe von SQL-Skripts in ein reales DBMS implementieren</li> <li>• Mit ausgewählten Testdaten und entsprechenden SQL-Befehlen die vorgegebenen Anforderungen an die Datenbank überprüfen</li> <li>• Integritätsbedingungen auf Datenbankbankebene umsetzen und dessen Wirkung beurteilen</li> </ul>	<ul style="list-style-type: none"> <li>• Vergleich mit imperativen und funktionalen Sprachen</li> </ul>
<p style="text-align: center;"><b>Benutzungsschnittstellen</b></p> <ul style="list-style-type: none"> <li>• einfache Kommando-interpretierer und grafische Oberflächen zur Administration</li> <li>• Zugriffe auf Datenbanken über Programmiersprachen z.B. PHP, Java</li> <li>• Datenbank-Web-Anbindungen</li> <li>• Ergonomie</li> </ul>	<ul style="list-style-type: none"> <li>• Entwicklung einer Benutzungsschnittstelle für den Datenbankzugriff über das Netz.</li> <li>• Implementierung von Zugriffsmöglichkeiten auf die Datenbank über das Netz.</li> <li>• Darstellung und Handhabung der Informationen aus ergonomischer Sicht</li> <li>• Integritätsbedingungen implementieren und deren Wirkung auf Client- und Serverseite beurteilen</li> </ul>	<ul style="list-style-type: none"> <li>• Designentscheidungen mit dem Fach <b>Kunst</b> absprechen.</li> <li>• Mehrsprachliche Benutzungsschnittstellen mit den jeweiligen <b>Fremdsprachen</b></li> </ul>

<b>Datenschutz und Datensicherheit in Datenbanksystemen</b>		
<ul style="list-style-type: none"> <li>• Steuerung von Zugriffsrechten und Benutzungssichten</li> <li>• Sperrmethoden (Record- und Filelocking)</li> <li>• Sicherheitsrisiken für Angriffe von Außen (z.B. Code Injection)</li> <li>• Sicherung der Integrität der Datenbank „im Betrieb“, (z. B. durch Indizierung der Schlüsselfelder)</li> <li>• Datenbank-Transaktionen, Wiederherstellung im Fehlerfall (Sicherheitskopien, Protokoll-Datei)</li> </ul>	<ul style="list-style-type: none"> <li>• Zugriffstabellen entwerfen und in SQL umsetzen.</li> <li>• Bei aktuellen DBMS die jeweiligen Sperrmethoden feststellen.</li> <li>• Konzepte zur Daten- und Transaktionssicherung beurteilen.</li> <li>• Grundsätzliche Sicherheitsrisiken für Angriffe kennen und Gegenmaßnahmen entwickeln.</li> </ul>	<ul style="list-style-type: none"> <li>• Vergleich mit den Manipulationsmöglichkeiten herkömmlicher Programme</li> <li>• Gefährdung von Informationssystemen (<b>Politikwissenschaften</b>)</li> </ul>

### Kommentierte Bücher- und Linkliste

#### *Literaturliste:*

- <http://www.bics.be.schule.de/inf2/datenbanken/literatur.html>

#### *Unterrichtsmaterialien und Online-Datenbanken:*

- <http://www.bics.be.schule.de/inf2/datenbanken/index.html>
- <http://oszhdl.be.schule.de/gymnasium/faecher/informatik/datenbanken/index.htm>



## F Computergrafik

Das Gebiet der grafischen Datenverarbeitung ist vom Umfang her derartig groß, dass eine sinnvolle Beschäftigung nach einem kurzen Überblick nur mit Teilaspekten möglich ist. Daher sollte nach einer Einführung entweder ein Gebiet aus dem Bereich der Pixel- oder der Vektorgrafik gewählt werden. Die wesentlichen Entwürfe von Datenobjekten und Zugriffsalgorithmen sollen in Eigenprogrammierung erledigt werden, zur Anzeige kann auf Bibliotheken oder fertige Software zurückgegriffen werden.

Computergrafik		
Vorschläge für Schwerpunktvorhaben:		
<ul style="list-style-type: none"> <li>• Unterschiede zwischen Pixelgrafik und Vektorgrafik</li> <li>• Alternative A: Pixelgrafik als das Rechnen mit rechteckigen Zahlenschemata</li> <li>• Alternative B: Vektorgrafik als sichtbar gemachte Geometrie</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<p><b>Einführung Pixelgrafik</b></p> <ul style="list-style-type: none"> <li>• Physik des Lichts, Physiologie des Auges, und der Farbwahrnehmung</li> <li>• Rasterung von Bildern, bei Aufnahme und Wiedergabe, Beziehungen zu Geräten</li> <li>• Rechnerinterne Darstellung von Pixelbildern in unkomprimierten Grafikformaten (Bitmap-, Graustufen- und Farbbildern)</li> <li>• Farbmodelle: RGB, CMY(K) mit Bezug zu Ausgabegeräten, additive Farbmischung für lichterzeugende Geräte (z.B. Monitore), subtraktive für reflexionsverändernde (z.B. Drucker)</li> </ul>	<ul style="list-style-type: none"> <li>• Identifizieren der Grundlagen in den Bezugswissenschaften</li> <li>• Grenzen der menschlichen Wahrnehmung als technische Rahmenbedingungen erkennen</li> <li>• Beschreibung des Rasterbildmodells über die Parameter Auflösung (horizontal und vertikal), Diskretisierung und Digitalisierung der Rasterpunktfarbwerte, technische Realisierung in Aufnahme- und Wiedergabegeräten</li> <li>• Bezüge zu anderen Digitalisierungsvorgängen herstellen</li> <li>• Einfache Grafiken durch Rechnung erzeugen, einfache Veränderungen an Pixelbildern vornehmen</li> <li>• Speicherplatzbedarf in Abhängigkeit von Auflösung und Farbtiefe ermitteln</li> <li>• Inner- und außerinformatische Farbmodelle und -mischmethoden zueinander in Beziehung setzen, ihre Anwendungsbereiche beschreiben</li> <li>• Verschiedene Modelle ineinander umrechnen</li> </ul>	<ul style="list-style-type: none"> <li>• Elektromagnetische Strahlung, Sonnenlicht, Auge (<b>Physik, Astronomie, Biologie</b>)</li> <li>• Mosaik, Pointilismus (<b>Kunstgeschichte</b>)</li> <li>• Musterweberei, Gobelins (<b>Kunsth Handwerk</b>)</li> <li>• Klassische Fotografie, Film (<b>Chemie</b>), analoges Fernsehen</li> <li>• Visualisierung von Rechenvorgängen</li> <li>• Datenmengen bei Speicherung und Transport durch Netze, Kompression und Formatwandlung</li> <li>• Farbenlehre (Bildende Kunst), Farbdruck (Grafisches Gewerbe)</li> <li>• Biochemische Vorgänge in Sehzellen</li> </ul>

<ul style="list-style-type: none"> <li>• Vergrößerung eines einfachen, zweidimensionalen geometrischen Objekts (um ganzzahligen Faktor)</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>• Zusammenhang zwischen Bild- und Informationsvergrößerung formulieren, Abschätzen der Wirkung rechnerischer Auflösungsvergrößerung bei Geräten (Interpolation)</li> <li>•</li> </ul>	
<p style="text-align: center;"><b>Vektorgrafik</b></p> <ul style="list-style-type: none"> <li>• Beziehungen zur analytischen und Abbildungsgeometrie</li> <li>• Koordinatensysteme: mathematisch (kartesisch) und speicherorientiert</li> <li>• Geometrische Objekte als Datenobjekte</li> <li>• Umwandlung in Pixelgrafik bei optischer Darstellung</li> <li>• Vergrößerung eines einfachen zweidimensionalen Objekts</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>• Vergleichen der mathematischen Grundelemente Punkt, Gerade, Ebene, Raum, Vektor mit den Elementen der Pixelgrafik</li> <li>• Aussageformen (Gleichungen) als Algorithmen interpretieren</li> <li>• Vergleichen der unterschiedlichen Zahlenmengen, die in Mathematik und Informatik z.B. das Koordinatensystem einer Ebene aufspannen</li> <li>• Erzeugen einfacher Zeichenobjekte in einer grafischen Oberfläche</li> <li>• Beziehungen zwischen Gleichungen/Koeffizienten und Objekten/Parametern erläutern.</li> <li>• Entwerfen von Klassen für einfache geometrische Objekte</li> <li>• Zeichnen der Objekte durch Markieren von Rasterpunkten für z.B. Geraden und Rechtecke</li> <li>• Anwenden mathematischer Abbildungsfunktionen, Vergleichen mit Pixelgrafikvergrößerung</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Geschichte der Geometrie (Mathematik)</b></li> <li>• Kartografie</li> <li>• Ältere mechanische Vergrößerungsgeräte (Storchenschnabel), Vergrößerungen bei optischen Geräten, Kopiergeräten</li> </ul>
<p style="text-align: center;"><b>Alternative A: Pixelgrafik</b></p> <ul style="list-style-type: none"> <li>• Linienalgorithmus und Treppenproblematik</li> <li>• Probleme der Skalierung von Pixelgrafiken, Anti-aliasing</li> <li>• Farbauszüge, Farbmodell HSV, Farbveränderungen</li> <li>• Veränderung von Helligkeit und Kontrast</li> </ul>	<ul style="list-style-type: none"> <li>• Programmieren des Bresenham- oder Mittelpunkt-Algorithmus</li> <li>• Programmierung durch Überdeckungsalgorithmus</li> <li>• Erstellen von 3- und 4-Farbauszügen, Falschfarbenbildern</li> <li>• Ermitteln von Helligkeitsverteilungen, Kontrastverstärkung und –vermin-</li> </ul>	<ul style="list-style-type: none"> <li>• Plotteralgorithmen ohne Multiplikation (<b>Rechnerorganisation</b>)</li> <li>• Kontrast und Auflösung bei Schärfewahrnehmung des Auges</li> <li>• Satellitenfotos, Infrarotfotografie</li> <li>• Analoge optische Geräte, Analogvideo, Fernsehen</li> </ul>

<ul style="list-style-type: none"> <li>• Transparenz und Mischen von Bildern</li> <li>• Umgebungsabhängige Algorithmen, Filterung, Konturfindung</li> <li>• Analyse von Bildbearbeitungsprogrammen</li> <li>• Anwendungen grafischer Datenverarbeitung in Medizin und Technik</li> </ul>	<p>derung</p> <ul style="list-style-type: none"> <li>• Erweitern der Bildklasse um einen Kanal für Transparenz und Bildmischoperation</li> <li>• Erstellung von Schärfungen und Unschärfen, Trennlinienermittlung</li> <li>• Identifizieren selbsterstellter Funktionen in Software, skizzieren möglicher Algorithmen für andere, nicht implementierte Funktionen</li> <li>• Studium ausgewählter Beispiele, Identifizieren der bearbeiteten Funktionen, Vergleichen</li> </ul>	<ul style="list-style-type: none"> <li>• Durchsichtige, durchscheinende und undurchsichtige Materialien, Transparenz bei Folien (<b>Physik</b>)</li> <li>• Optische Zeichenerkennung, Luftbildanalyse und Kartografie</li> <li>• Vergleichen der betrachteten Software mit der Arbeitsweise im Einsatzgebiet vor dem Rechnereinsatz</li> </ul>
<p><b>Alternative B: Vektorgrafik</b></p> <ul style="list-style-type: none"> <li>• Grundlagen vektorieller Geometrie</li> <li>• Klassen einfacher zweidimensionaler geometrischer Objekte</li> <li>• Operationen: Skalieren, Verschieben, Drehen, Spiegeln, Füllen</li> <li>• Übertragung auf einfache dreidimensionale Objekte (Drahtmodelle)</li> <li>• Projektion auf eine Ebene</li> <li>• Operationen: Skalieren, Verschieben, Drehen</li> <li>• Einfache Animation</li> <li>• Analyse nichtproprietärer Formate: z.B. SVG, VRML</li> <li>• Anwendungen in Präsentationen, Werbung, industrieller Konstruktion</li> </ul>	<ul style="list-style-type: none"> <li>• Einfache geradlinige Objekte mathematisch beschreiben können, Schnittpunkte von Linien berechnen</li> <li>• Programmieren und testen, Methoden für Schnittpunkte und Innenpunktbestimmung, Vererbung benutzen</li> <li>• Mathematische Operationen umsetzen, Effizienzüberlegungen anstellen</li> <li>• Klassen erzeugen für Geraden, Strecken, Ebenen</li> <li>• Erzeugen einer einfachen Projektion für die Bildschirmdarstellung</li> <li>• Mathematische Operationen umsetzen, Effizienzüberlegungen anstellen</li> <li>• Vergleichen mit den Operationen im zweidimensionalen Raum</li> <li>• Erzeugen eines berechneten Bewegungsablaufs, speichern einzelner Bilder, in animiertes Grafikformat durch externe Software umsetzen</li> <li>• Programmieren eines beispielhaften Formatwandlers</li> <li>• Studium ausgewählter Beispiele, Identifizieren der bearbeiteten Funktionen,</li> </ul>	<ul style="list-style-type: none"> <li>• Geschichte der Geometrie (<b>Mathematik</b>)</li> <li>• Polarkoordinaten</li> <li>• Räumliche Darstellung und Perspektive (<b>Kunst, technische Zeichnung</b>)</li> <li>• Trägheit des Auges, Bildwechselfrequenzen bei Monitoren</li> <li>• Öffentliche und proprietäre Dateiformate, wirtschaftliche Interessen, Softwarepatente</li> <li>• Vergleichen der betrachteten Software mit der Arbeitsweise im Einsatzgebiet</li> </ul>

industrieller Konstruktion (CAD) und Simulation, Spielen und rechnergenerierten Filmteilen	Vergleichen	vor dem Rechnereinsatz
--	-------------	------------------------

### **Kommentierte Bücher- und Linkliste**

- **Foley, James D. et al.:** *Computer Graphics: Principles and Practice*  
ISBN: 0201357178, Jahr: 2003  
Die angekündigte Neuauflage dieses Standardwerkes ist Anfang 2005 noch nicht erschienen, die vorige Ausgabe sollte in jedem Fall noch zu bekommen sein. Eine durchaus noch brauchbare deutsche Übersetzung bei Addison-Wesley von 1994 ist gut zu benutzen, aber leider vergriffen. Das Werk enthält alles, was man über Pixelgrafik, Farben, Geräte, 2-D, 3-D, anspruchsvolle Anwendungen wie Raytracing etc. wissen können will, die meisten Algorithmen sind abgedruckt, es gibt dort reichlich Hilfe für weitere Arbeitsmöglichkeiten.
- **LOG IN Themenheft 5-6/2001** „Digitale Bilderwelten“. LOG IN Verlag Berlin.  
Einzelhefte erhältlich über <http://www.log-in-verlag.de>  
Das Heft enthält eine Reihe von Aufsätzen zum Thema Grafik sowie Unterrichtsbeispiele in JAVA und PYTHON mit Angaben für verwendbare Klassen als Quelltexte. Dort sind viele weitere Quellen für Literatur und Online-Recherche genannt. Online-Quellen sind auch über den Verlagsservice auf dessen WEB-Seiten zu erfahren.

## G Netzwerke

Die Technik und die Verwendung von elektronischen Datennetzen ist im Rahmen der Informatik ein Teilgebiet, das starken aktuellen Entwicklungen unterworfen ist. Eine Abgrenzung und Beschränkung der im Schulunterricht zu behandelnden Inhalte ist hier besonders schwierig. Von den unten genannten inhaltlichen Schwerpunkten muss im Hinblick auf den zur Verfügung stehenden Zeitumfang eine Auswahl getroffen werden.

Aspekte der Datensicherheit und des Datenschutzes, aber auch rechtliche und gesellschaftliche Fragen werden unter Umständen schon im vorangegangenen Unterricht thematisiert worden sein, müssen dann also in diesem Vertiefungsgebiet lediglich spezifisch ergänzt werden.

Zentral sollte vermittelt werden, wie die physikalisch-technischen Gegebenheiten die praktische Realisierung von Netzwerken aus informatischer Sicht bestimmen. Die Verwendung eines Netzwerkes rein aus Anwendersicht, z.B. die Erstellung eines Webprojektes, ist hingegen kein wesentlicher Teil dieses Wahlgebietes.

Die Kommunikation zwischen Rechnern basiert auf dem ISO/OSI-Referenzmodell, das für die Behandlung des Themas Rechnernetze einen übergeordneten Rahmen liefert. Es ist empfehlenswert die Anzahl der Schichten auf vier zu begrenzen (Anwendungsschicht, Transportschicht, Netzwerkschicht und physikalische Schicht). Die aufgeführten Inhalte sollten aus der Sicht des schulischen Netzwerkes behandelt werden.

Ein möglicher Einstieg in das Thema ist z.B. die Verwendung eines Werkzeuges, das Netzwerkverkehr mitschneidet und diesen analysieren kann (siehe Literaturhinweise). Ein möglicher Abschluss des Projektes ist eine „LAN-Party“, in der ein lokales Netzwerk aufgebaut wird. Ein solches Projekt verbindet die erworbene Technikkompetenz mit der Lebenswirklichkeit von Schülern.

Computer-Netzwerke		
<b>Vorschläge für Schwerpunktvorhaben:</b> <ul style="list-style-type: none"> <li>• Ziele, Anforderungen und Einsatzgebiete von vernetzten Systemen</li> <li>• Netzwerktechnik, Hardware, physikalische Komponenten</li> <li>• Protokolle und Dienste</li> <li>• Datenschutz und Datensicherheit</li> <li>• Rechtliche und gesellschaftliche Aspekte</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<p style="text-align: center;"><b>Grundlagen</b></p> <ul style="list-style-type: none"> <li>• Analyse des Programms ELIZA als sprachverarbeitendes System</li> <li>• Ziele und Aufgaben von Netzen <ul style="list-style-type: none"> <li>- gemeinsamer Zugriff auf Daten</li> <li>- gemeinsame Nutzung von Hard- und Software</li> <li>- Lastverbund</li> <li>- Verfügbarkeitsverbund</li> </ul> </li> <li>• Klassifizierung nach Übertragungsweiten, Netztopologien (GAN, WAN, MAN, LAN)</li> </ul>	<ul style="list-style-type: none"> <li>• Anforderung an Netze klassifizieren und beurteilen</li> <li>• Aufgaben in ihrer Vielfalt kennen und am eigenen Schulnetz bewusst machen</li> <li>• Netze einordnen (z. B. Schulnetz, LAN-Party-Netz)</li> <li>• Entwicklung von Computernetzen im historischen Bezug darstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Internetorientierter Zugang im ersten Jahr Informatikunterricht</li> <li>• Einordnung in die geschichtliche Entwicklung der Informatik (erstes Unterrichtsjahr)</li> </ul>

<ul style="list-style-type: none"> <li>• Geschichte der Vernetzung am Beispiel des Internet</li> </ul>		
<p style="text-align: center;"><b>Datenübertragung</b></p> <ul style="list-style-type: none"> <li>• Aufbau und Komponenten von Datennetzen an ausgewählten Beispielen, z.B. <ul style="list-style-type: none"> <li>- Ethernet</li> <li>- Tokenring</li> <li>- WLAN (Funknetze)</li> <li>- Telefon/ISDN</li> <li>- Mobilfunk (GSM und UMTS)</li> <li>- WAN/Internet</li> </ul> </li> <li>• Übertragungsmedien <ul style="list-style-type: none"> <li>- BNC-Kabel</li> <li>- RJ45</li> <li>- Lichtwellenleiter</li> <li>- Funk)</li> </ul> </li> <li>• Hardwarekomponenten <ul style="list-style-type: none"> <li>- Hub</li> <li>- Switch</li> <li>- Router</li> </ul> </li> <li>• Übertragungstechniken <ul style="list-style-type: none"> <li>- Paket- und</li> <li>- Leitungsvermittlung)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Struktur von Datennetzen erläutern</li> <li>• Unterschiede und Einsatzgebiete der verschiedenen Übertragungsmedien und Hardwarekomponenten kennen</li> <li>• die Vorteile der paketorientierten Vermittlung am Beispiel Internet erläutern</li> <li>• die technologische Entwicklung von Rechnernetzen erkennen und abschätzen</li> </ul>	<ul style="list-style-type: none"> <li>• Technische Informatik, Rechnerstrukturen</li> <li>• Codierung, Fehlererkennung, Kompressionsverfahren</li> </ul>
<p style="text-align: center;"><b>TCP/IP Netzwerke</b></p> <ul style="list-style-type: none"> <li>• IP</li> <li>• Adressierung und Pakete</li> <li>• Routing und Netzwerkklassen</li> <li>• TCP</li> <li>• Evtl. Ausblick auf IPv6</li> </ul>	<ul style="list-style-type: none"> <li>• Bedeutung von Protokollen</li> <li>• Aufbau und Eigenschaften von IP und TCP beschreiben</li> <li>• Funktion und Aufgaben von Routing und Netzwerkklassen benennen</li> </ul>	<ul style="list-style-type: none"> <li>• Graphentheorie (Routing-Algorithmen, <b>Mathematik</b>)</li> </ul>
<p style="text-align: center;"><b>Dienste und Anwendungen</b></p> <ul style="list-style-type: none"> <li>• DNS</li> <li>• Telnet, FTP</li> <li>• HTTP</li> <li>• E-Mail (SMTP, POP3)</li> <li>• Kommunikation über Ports und Sockets</li> </ul>	<ul style="list-style-type: none"> <li>• Zusammenhang zwischen Domain, IP-Adresse und physikalischer Adresse herausarbeiten und verstehen</li> <li>• Dienste des Internet kennen und abgrenzen</li> </ul>	<ul style="list-style-type: none"> <li>• Sicherheitsfragen beim Datenaustausch</li> <li>• Programmierung von verteilten Anwendungen (Projektsemester)</li> </ul>
<p style="text-align: center;"><b>Datenschutz und Datensicherheit</b></p> <ul style="list-style-type: none"> <li>• Sicherheitsrisiken <ul style="list-style-type: none"> <li>- Viren</li> <li>- Paketschnüffler</li> <li>- aktive Inhalte</li> <li>- Würmer</li> <li>- Trojaner.</li> </ul> </li> <li>• Schutzmaßnahmen <ul style="list-style-type: none"> <li>- Identitätskontrolle</li> <li>- Authentizitätskontrolle</li> </ul> </li> <li>• Verschlüsselung</li> <li>• Firewalls</li> </ul>	<ul style="list-style-type: none"> <li>• Sicherheitsrisiken in Netzen (z. B. LAN, Internet) kennen und beurteilen</li> <li>• Schutzmaßnahmen anwenden</li> <li>• Informationsquellen über Gefahren und Abwehrmöglichkeiten kennen und bewerten</li> </ul>	<ul style="list-style-type: none"> <li>• Kryptologie</li> <li>• Bezüge zum Zeitgeschehen</li> </ul>

Rechtliche und gesellschaftliche Aspekte		
<ul style="list-style-type: none"> <li>• Rechtliche Grundlagen                             <ul style="list-style-type: none"> <li>- Urheberrecht</li> <li>- Telekommunikationsrecht</li> <li>- informationelle Selbstbestimmung (Datenschutzrecht)</li> </ul> </li> <li>• Gesellschaftliche Auswirkungen in den Bereichen Arbeitswelt, Wirtschaft und Verwaltung</li> </ul>	<ul style="list-style-type: none"> <li>• Rechtlichen Grundlagen überblicken und beachten</li> <li>• gesellschaftliche Auswirkungen exemplarisch untersuchen und beurteilen</li> </ul>	<ul style="list-style-type: none"> <li>• Zu gesellschaftlichen Diskussionen können Bezüge mit Hilfe aktueller Quellen hergestellt werden (<b>Politische Weltkunde</b>)</li> <li>• Neue Wettbewerbsstrukturen, z. B. im mobilen Telefonieren und Datenverkehr (<b>Wirtschaftslehre</b>)</li> </ul>

## Kommentierte Bücher- und Linkliste

### Bücher

- **Gumm, H. P.:** *Einführung in die Informatik*. Oldenbourg Verlag. München 2000.  
Viele Grundbegriffe der Informatik werden in verständlichen Worten erläutert. Es umfasst einen Abschnitt über das Internet, der die wichtigsten Dienste wie E-Mail, FTP, World Wide Web (WWW), Gopher und News vorstellt.
- **Kurose, J. F.; Ross, K. W.:** *Computernetze*, Bafög Ausgabe. Pearson Studium. 2004.  
Die Autoren gehen in ihrem Buch das Thema Computernetzwerke aus einer neuen Perspektive an. Sie verwenden einen modernen Top-Down-Ansatz, der mit dem am stärksten wachsenden Gebiet der Netzwerkentwicklung, der Anwendungsebene beginnt und dann dem Protokollstapel folgt. Durch das ganze Buch hinweg demonstrieren Beispiele aus der Internetarchitektur, wie die theoretischen Grundlagen in die Praxis umgesetzt werden. Auf der Companion Website sind zusätzliche Materialien für Studenten inklusive interaktivem Lernmaterial und viele interessante Links zu finden - für Dozenten in einem gesonderten Bereich Power-Point-Folien, Laborprojekte und Lösungen zu den Übungen des Buches.
- **Medosch, Röttgers:** *Netzpiraten - Die Kultur des elektronischen Verbrechens*. Heidelberg: dpunkt-Verlag 2001.  
„Netzpiraten“ informiert fundiert über Themen wie Viren, Hoaxes oder Spam. In dem Buch wird versucht, Subkulturen (Hacker, Cracker und Raubkopierer) des Internets vorurteilsfrei zu beschreiben und die Rechtslage übersichtlich aufzuzeigen.
- **Peterson, L. , Davie B.:** *Computernetze*. Dpunkt Verlag. 2000.  
Das Buch liefert ausführliche Hintergrundinformationen zur Arbeitsweise von Netzwerken und Protokollen. Der Schwerpunkt liegt dabei bei IP und Internet.
- **Stein, E.:** *Taschenbuch Rechnernetze und Internet*, 2. Auflage. Fachbuchverlag Leipzig, 2001.  
Das Taschenbuch gibt einen umfassenden, systematischen Überblick zum Gesamtgebiet Rechnernetze und Internet.
- **Tanenbaum A. S.:** *Computer-Netzwerke*. 4. Auflage (auf Deutsch). Pearson Studium 2003.  
Tanenbaums „Computernetzwerke“ ist wahrscheinlich das weltweit bekannteste und verbreitetste Lehrbuch zum Thema Rechnernetze. Die vorliegende 4. Auflage behandelt auch aktuelle Themen wie ADSL, drahtloses Breitband, das Internet, das Ethernet und drahtlose LANs. Ein komplett neues Kapitel ist dem Thema Sicherheit gewidmet.

### Links

- **<http://www.ontolearn.de/moodle/login/index.php>**  
Online-Kurs zum Thema Computernetzwerke von Andreas Rittershofer (z. Z. noch im Aufbau)
- **<http://www.internet-kompetenz.ch/>**  
Selbststudienmaterial zum Aufbau des Internet (incl. Dienste) mit praktischen Übungen von Michael Näf, ETH Zürich. Die Plattform bietet Lernmaterialien, Literatur, Übungen und Kurse zum Thema Internet (Funktionsweise, Informationsbeschaffung, Sicherheit) an. Der Grundsatz:

Das Material soll sich auf die wichtigen grundlegenden Prinzipien konzentrieren und nicht durch unnötiges Detailwissen Verwirrung stiften.

- **<http://www.www-kurs.de/>**  
Auf diesem Portal finden sich neben Erklärungen zur Historie und Struktur des Internets auch sehr viel weitergehende Informationen rund um das Arbeiten mit und im Netz, sowie ein recht umfangreiches Glossar.
- **<http://www.educeth.ch/informatik/gruppen/firewall/index.html>**  
In einer Unterrichtseinheit wird im ersten Teil auf das Konzept und die Funktionalität einer Firewall eingegangen. Im zweiten Teil liegt der Fokus auf der technischen Ebene. Basierend auf den unterschiedlichen Firewalltypen werden verschiedene Architekturen und Aspekte der Implementierung diskutiert.
- **<http://www.educeth.ch/informatik/lernaufg/routing/>**  
Umfangreiche Unterrichtsmaterialien zu Thema "Internet-Routing" von Patrick Seule, ETH Zürich. Verschiedene Routing-Algorithmen werden mit so genannten Lernaufgaben hergeleitet und bezüglich ihrer Praxisbezuges analysiert.
- **<http://www.bildung.hessen.de/abereich/inform/skii/material/delphi/internet.htm>**  
Auf dem hessischen Bildungsserver wird einer Unterrichtssequenz zu den Grundlagen des Internet dargestellt. Anhand einiger Beispiele wird gezeigt, wie Rechnerkommunikation auf der Basis von Internetprotokollen realisiert wird.
- **<http://www.net.informatik.tu-muenchen.de/teaching/former/jessenScripts.html>**  
Sehr umfangreiche Vorlesungsskripte "Rechnernetze" von Prof. Jessen (TU München).
- **<http://einstein.informatik.uni-oldenburg.de/rechnernetze/Default.htm>**  
Online-Vorlesungsskript von Prof. Dr. W. Kowalk (Universität Oldenburg)
- **<http://www.netplanet.org/dienste/>**  
Ein übersichtliches Online-Lexikon zum Thema Internet (Adressierung, Aufbau, Dienste, Geschichte, Sicherheit, Netiquette)
- **<http://www.lehrer-online.de/url/lan-party>**  
Technische und rechtliche Anmerkungen zur Organisation einer LAN-Party in der Schule.

## H Künstliche Intelligenz (KI)

„Können Maschinen denken?“, „Sind Menschen Maschinen?“, „Ist künstliches Leben möglich?“ sind grundlegende Fragen, die Menschen schon immer interessiert haben. In diesem Vertiefungsgebiet können solche Fragen auf dem Hintergrund der Ergebnisse der Forschung zur künstlichen Intelligenz (KI) und zum künstlichen Leben (KL) reflektiert werden. KI/KL ist ein wichtiges Teilgebiet der Informatik, deren Vertreter in den letzten Jahrzehnten häufig durch übertriebene Ankündigungen und Fehlschläge in der praktischen Umsetzung auf sich aufmerksam gemacht haben. Trotzdem finden sich Forschungsergebnisse aus KI/KL schon heute in vielen Systemen wieder: sprachverarbeitende Computer („Chatbots“), Expertensysteme z. B. in der Medizin, fußballspielende Roboter, neuronale Netze zur Mustererkennung usw.

Im Folgenden wird exemplarisch ein Unterrichtsgang beschrieben, der sich auf ein Problem der „klassischen“ KI konzentriert: die „Unterhaltung“ mit einem Computer. Diese Idee ist Grundlage des Turing-Tests zur Definition von Maschinen-Intelligenz. Da heutzutage die meisten Anrufe bei Firmen durch „Chatbots“ beantwortet werden ist dieses Thema aber nicht nur von philosophischem sondern auch von praktischem Interesse.

Weitere interessante Themen im Bereich KI/KL sind möglich und sinnvoll, z. B.

- Viren, Würmer und andere „Lebewesen“ in Computern und Netzen
- Evolution und Selbstreproduktion im Computer
- Künstliches Leben und Robotik
- Computer und Gehirn: Von-Neumann-Rechner vs. Neuronales Netz

Themen aus dem Gebiet KI/KL haben zahlreiche Verbindungen zum Unterricht anderer Fächer (Biologie, insbes. Hirnforschung und Bioinformatik, Religion und Ethik, Literatur und Sprachwissenschaft) und bieten sich daher in besonderer Weise für fächerübergreifenden und fächerverbindenden Unterricht, z.B. in einem Seminarkurs, an.

Künstliche Intelligenz (KI)		
<b>Vorschläge für Schwerpunktvorhaben:</b> <ul style="list-style-type: none"> <li>• Was sind und wie arbeiten Chatbots?</li> <li>• Möglichkeiten und Grenzen maschineller Sprachverarbeitung</li> <li>• Der Turing-Test</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<b>ELIZA von J. Weizenbaum</b> <ul style="list-style-type: none"> <li>• Analyse des Programms ELIZA als sprachverarbeitendes System</li> </ul>	<ul style="list-style-type: none"> <li>• Pattern-Matching als grundlegendes Prinzip von ELIZA erkennen</li> <li>• Syntax und Semantik unterscheiden</li> <li>• Wirkungsgeschichte des Eliza-Programms kennen</li> <li>• Einfache eigene Skripte erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Beschreibung der „Wissensbasis“ (Skript) von ELIZA durch formale Sprache</li> <li>• Probleme bei der Generierung grammatikalisch korrekter Antworten (<b>Deutsch</b>)</li> </ul>
<b>Der Turing-Test</b> <ul style="list-style-type: none"> <li>• Der Turing-Test als Möglichkeit zur Definition von künstlicher Intelligenz</li> </ul>	<ul style="list-style-type: none"> <li>• Die Definition des Turing-Tests kennen und beurteilen</li> <li>• Grundlegende Kritik an der Intelligenz-Definition von Turing durchdenken (z. B. das „chinesische Zimmer“)</li> </ul>	<ul style="list-style-type: none"> <li>• Leib-Seele-Problem (<b>Philosophie</b>)</li> </ul>

	<ul style="list-style-type: none"> <li>• Aufklärung über praktische Probleme bei der automatischen Sprachverarbeitung verschaffen (z. B. „allgemeines Hintergrundwissen“)</li> </ul>	
<p><b>Maschinelle Sprachverarbeitung</b></p> <ul style="list-style-type: none"> <li>• Moderne sprachverarbeitende Computer („Chatbots“)</li> </ul>	<ul style="list-style-type: none"> <li>• Experimente mit aktuellen Chatbots durchführen</li> <li>• Die Wissensbasis der Chatbots einschätzen und ggf. erweitern</li> </ul>	<ul style="list-style-type: none"> <li>• Loebner-Preis für Programme, die den Turing-Test möglichst gut erfüllen</li> <li>• Untersuchungen z. B. zu ALICE (mehrfache Gewinnerin des Loebner-Preises) und der dabei verwendeten Sprache AIML</li> </ul>

### **Kommentierte Bücher- und Linkliste**

Diese Listen sowie Materialien aus Fortbildungen von H. Witten finden sich z. Zt. unter <http://bscw.schule.de/pub/bscw.cgi/0/264020>, später werden sie auch auf dem Berliner Bildungsserver zu finden sein.

## I Technische Informatik

Im vorangegangenen Informatikunterricht blieb der Bezug zur Rechnertechnik auf Grundstrukturen der Rechenanlage aus Sicht des Anwenders beschränkt. Nun wird der Blick in das Innere der Rechenanlagen in logischer (Bausteine) wie semantischer (Betriebssysteme) Hinsicht erweitert. Die Beschreibung des Systems durch Zustandsgraphen bleibt unberücksichtigt. Die Assembler-Programmierung zählt zum Wahlthema K (Maschinennahe Programmierung).

Technische Informatik		
Vorschläge für Schwerpunktvorhaben:		
<ul style="list-style-type: none"> <li>• Funktionsweise von Computerkomponenten an ausgewählten Beispielen der Digitaltechnik</li> <li>• Einblick in Prinzipien der Rechnerarchitektur</li> <li>• Einordnung von Betriebssystemen</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<p style="text-align: center;"><b>Strukturen der Rechnerarchitektur</b></p> <ul style="list-style-type: none"> <li>• Grundsätzliche Strukturen nach Zuse, von Neumann (Harvard-Variante), und auch Parallelrechner</li> <li>• Endlichkeit der technisch realen Maschine hinsichtlich Speicherraum und Geschwindigkeit</li> <li>• Bussysteme (Datenbus, Adressbus, Steuerbus)</li> </ul>	<ul style="list-style-type: none"> <li>• Unterscheidungsmerkmale zur Klassifizierung von Rechnerarchitektur kennen</li> <li>• Ursachen anwendungsbezogener Probleme beurteilen</li> <li>• Durch die vertiefte historische und ökonomische Einordnung ein erweitertes Verständnis für die fortschreitende Entwicklung von Rechenanlagen erlangen</li> </ul>	<ul style="list-style-type: none"> <li>• Sozio-ökonomische Bedingungen (<b>Geschichte, Politische Weltkunde</b>)</li> <li>• Verbindung zur Thematik „Verlässlichkeit der Informationstechnik“</li> <li>• Vergleich mit bekannten Systemen, Vorerfahrungen der Schülerinnen und Schüler</li> </ul>
<p style="text-align: center;"><b>Logische Funktionen</b></p> <ul style="list-style-type: none"> <li>• Grundfunktionen AND, OR, NOT</li> <li>• Kombinationen der Grundfunktionen zu einfachen FlipFlop- Speicherbausteinen und Volladdierern</li> <li>• Schieberegister</li> <li>• AD/DA Wandler</li> </ul>	<ul style="list-style-type: none"> <li>• CPU- Struktur und ihre Wirkungsweise verstehen</li> <li>• Am Beispiel der Schieberegister die seriell / parallel Umsetzungen erläutern</li> <li>• Zugang zur Analyse prozesstechnischer Anwendungen finden</li> </ul>	<ul style="list-style-type: none"> <li>• Die elektrischen Grundfunktionen können ggf. mit Bausteinen des <b>Physik</b>-sammlungen praktisch untersucht werden</li> </ul>
<p style="text-align: center;"><b>Zahlensysteme</b></p> <ul style="list-style-type: none"> <li>• Begriff der Stellenwertigkeit</li> <li>• Zahlendarstellung binär, sedezimal und dezimal</li> <li>• Regeln für die Umrechnungen</li> </ul>	<ul style="list-style-type: none"> <li>• Technologische Entwicklungen anhand von Codierungen beschreiben und bewerten</li> <li>• Die Verlässlichkeit realer technischer Systeme durch Vergleiche numerischer</li> </ul>	<ul style="list-style-type: none"> <li>• Die Umwandlungsalgorithmen stellen (nur geringe) Anforderungen an die <b>Mathematik</b></li> </ul>

<ul style="list-style-type: none"> <li>• Regeln für das Addieren (Subtrahieren) von Zahlen in dualer Darstellung</li> <li>• Repräsentation von Gleitkommazahlen und negativen ganzen Zahlen in dualer Darstellung</li> </ul>	<p>Ergebnisse mit verifizierten Testdaten erfahren</p>	
<p style="text-align: center;"><b>Betriebssysteme</b></p> <ul style="list-style-type: none"> <li>• Betriebssystemarten (single- und multitasking, single- und multiuser)</li> <li>• Begriff der Echtzeit</li> <li>• Prozess, Task, Thread</li> </ul>	<ul style="list-style-type: none"> <li>• Verschiedene Systeme der Prozesssteuerung an Simulationen beobachten und beschreiben</li> <li>• Betriebssysteme hinsichtlich der Einsatzanforderungen unterscheiden und bewerten</li> <li>• Praktische Installationsübungen an isolierten Rechnersystemen durchführen</li> </ul>	<ul style="list-style-type: none"> <li>• Strategien der Prozesssteuerung können in Rollenspielen veranschaulicht werden (Round Robin, Speisende Philosophen, Erzeuger- Verbraucher, usw.)</li> <li>• Je nach vorher eingeführter Programmiersprache sind kleine Übungsbeispiele zu nichtsequentieller Programmierung hilfreich</li> </ul>

### Kommentierte Bücher- und Linkliste

#### Schulbuch

**Modrow, E.:** *Technische Informatik mit Delphi*, emu-online Scheden, 2004, 3-8334-1501-0

#### Links

##### Rechnerarchitektur

- <http://ei.cs.vt.edu/~history/>
- <http://www.netzmafia.de/skripten/bs/>
- [http://www.cs.vu.nl/~ast/books/book\\_software.html](http://www.cs.vu.nl/~ast/books/book_software.html)

##### Digitaltechnik

- [https://sourceforge.net/project/showfiles.php?group\\_id=15930](https://sourceforge.net/project/showfiles.php?group_id=15930)

##### Zahlensysteme

- <http://people.freenet.de/j.kubiak/rechnendual.pdf>
- <http://www.iam.fh-flensburg.de/MatheProp/Vorlesungen/Kap3/>
- <http://pi1.physi.uni-heidelberg.de/physi/ausbildung/andiweidner/berichte/zahlensysteme.pdf>
- [http://public.rz.fh-wolfenbuettel.de/~maartenh/IG1\\_WS0203/Kapitel7x4.pdf](http://public.rz.fh-wolfenbuettel.de/~maartenh/IG1_WS0203/Kapitel7x4.pdf)

## K Maschinennahe Programmierung

Die technischen Anwendungen der Informatik sind oft weiter verbreitet, als Schüler annehmen. Fast alle ursprünglich analog arbeitenden Lösungen werden heute mit Hilfe kleiner Prozessoren digital realisiert, z. T. mit sogenannten „embedded systems“. Die meisten Prozessoren finden sich heute nicht in PCs, sondern in anderen technischen Geräten (je nach Quelle 80 % bis über 90 %).

Mit Hilfe der Programmierung von Mikroprozessor-Minimalsystemen – wie sie als Realobjekte preisgünstig angeboten werden (z.B. C-Control), – oder der Programmierung simulierter Systeme (z. B. [www.digitalsimulator.de](http://www.digitalsimulator.de)) können die Grundlagen der verwendeten Konzepte vermittelt werden. Besonders geeignet erscheinen auf das Wesentliche reduzierte „didaktische Prozessorsimulationen“ oder ältere 8- bis 16- Bit Real-Prozessoren. Das von A. K. Dewdney 1984 vorgeschlagene „Core War“ eignet sich für eine motivierende Einführung in die maschinennahe Programmierung.

Maschinennahe Programmierung		
Vorschläge für Schwerpunktvorhaben: <ul style="list-style-type: none"> <li>• Einblick in die Rechnerarchitektur</li> <li>• Grundlagen der Assemblerprogrammierung</li> <li>• Digitale Kommunikation zur Erfassung von Peripheriezuständen und zur programmgesteuerten Beeinflussung externer Systeme</li> </ul>		
Fachinhalte	Kompetenzen	Vernetzungen
<p style="text-align: center;"><b>Einblick in die Rechnerarchitektur</b></p> <ul style="list-style-type: none"> <li>• grundsätzliche Strukturen nach Zuse, von Neumann (Harvard-Variante), zum Vergleich auch Parallelrechner</li> <li>• Bussysteme (Datenbus, Adressbus, Steuerbus)</li> </ul>	<ul style="list-style-type: none"> <li>• Das Informieren über interne Abläufe erschließt neben Unterscheidungsmerkmalen auch Ursachen anwendungsbezogener Probleme</li> <li>• Durch die vertiefte historische und ökonomische Orientierung soll ein erweitertes Verständnis für die fortschreitende Entwicklung von Rechanlagen erlangt werden</li> </ul>	<ul style="list-style-type: none"> <li>• Sozio-ökonomische Bedingungen (<b>Geschichte, Politische Weltkunde</b>)</li> <li>• Vergleich mit bekannten Systemen und Vorerfahrungen der Schülerinnen und Schüler</li> </ul>
<p style="text-align: center;"><b>Grundlagen der Assemblerprogrammierung</b></p> <ul style="list-style-type: none"> <li>• Zeitverhalten und logisches Verhalten digitaler Bauelemente</li> <li>• Begriff der Stellenwertigkeit</li> <li>• Zahlendarstellung binär, sedezimal und dezimal</li> <li>• Regeln für die Umrechnungen</li> <li>• Regeln für das Addieren (Subtrahieren) von Zahlen in dualer Darstellung</li> </ul>	<ul style="list-style-type: none"> <li>• Reflektieren, welche physikalischen Vorgänge eine rein logische Beschreibung der Bauelemente verbieten bzw. Validierung erfordern</li> <li>• Bewerten verschiedener Stände technologischer Weiterentwicklungen bei Kenntnis möglicher Codierungen</li> <li>• Anwenden und Untersuchen der realen Repräsentation der verschiedenen Daten im Arbeitsspeicher</li> </ul>	<ul style="list-style-type: none"> <li>• Bezug zu RC-Gliedern in der <b>Physik</b> (Zeitkonstanten) und Aussagenlogik (<b>Mathematik</b>)</li> <li>• Die Umwandlungsalgorithmen stellen einfache Anforderungen an die <b>Mathematik</b></li> </ul>

<p>in dualer Darstellung</p> <ul style="list-style-type: none"> <li>• Analog-Digitalwandlung, Digital-Analogwandlung</li> <li>• Einfache Beispiele der Maschinen-Programmierung (Speicherzugriffe, elementare Operationen)</li> <li>• Unterschiedliche Adressierungsarten bei der Assembler-Programmierung</li> <li>• Unterprogramme und Rückkehrstapel</li> <li>• Ein- und Ausgabe-Programmierung (Polling, bedingte Sprünge)</li> <li>• Ein- Ausgabe-Programmierung in Abhängigkeit von externen Ereignissen (Interruptverarbeitung)</li> </ul>	<ul style="list-style-type: none"> <li>• Mindestens jeweils ein Verfahren der Wandlung in Anwendungssituationen beurteilen und implementieren</li> <li>• Einfache Programme mit bis zu etwa zwanzig Instruktionen entwickeln</li> <li>• Typische Beispiele für Assemblerprogrammierung analysieren, erstellen und bewerten und dabei Prinzipien der zeitkritischen Programmierung und der Ansteuerung von Hardwareschnittstellen (Interfaces) beachten</li> <li>• Vorhandene Assemblerprogramme mit z.B. einer Alarm - Zusatzfunktion versehen</li> </ul>	<ul style="list-style-type: none"> <li>• Grundlagen elektronischer Schaltungen (<b>Physik</b>), Anwendung der Verfahren im Alltag (Telefon, CD usw.)</li> <li>• Reflexion entsprechender Konzepte bei höheren Programmiersprachen (Variablen, Datentypen, Operatoren / einfache Variablen, Arrays, Objektreferenzen (Pointer))</li> <li>• wissenschaftlich-technische Auswertungen in Echtzeit, (z.B. FFT), Fourier-Synthese und -Analyse (<b>Physik, Mathematik</b>)</li> <li>• Reflexion entsprechender Konzepte bei höheren Programmiersprachen (z.B. Behandlung von Mausereignissen)</li> </ul>
<p><b>Digitale Kommunikation zur Erfassung von Peripheriezuständen und zur programmgesteuerten Beeinflussung externer Systeme</b></p> <ul style="list-style-type: none"> <li>• Programmierung eines Datenloggers als Anwendungsbeispiel für: <ul style="list-style-type: none"> <li>- Temperaturmessungen</li> <li>- Anzeigetafeln</li> <li>- Temperatursteuerungen</li> <li>- Fahrzeuglogik</li> <li>- Alarmanlagen</li> <li>- Robotersteuerung</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Implementieren, Testen und Anwenden von Steuerprogrammen für Realanwendungen bei Beachtung vorgegebener Schnittstellspezifikation in Maschinsprache oder einer geeigneten Prozess-Programmiersprache</li> </ul>	<ul style="list-style-type: none"> <li>• Fächerübergreifende Aspekte ergeben sich durch die entsprechenden Prozessgrößen der <b>Biologie, Physik, Chemie</b> oder auch des <b>Sports</b> und allen Bereichen der <b>Technik</b></li> </ul>

## Kommentierte Bücher- und Linkliste

### Bücher

- **Link, W.:** *Assemblerprogrammierung*. Franzis-Verlag, ISBN: 3772375081
- **Backer, R.:** *Assembler - Maschinennahes Programmieren von Anfang an* 1. Auflage 2003, Rowohlt, ISBN: 3499612240
- **Hannemann:** *Mikroinformatik I und II*. Band 1: ISBN 3-920088-11-5, Band 2: ISBN 3-920088-20-4
- **Kainka, B.; Helbig, A.:** *Messen Steuern Regeln mit C-Control II*. Franzis-Verlag 2003
- **Domnick, Karl H.:** *Das 80C166er Lehrbuch*. Elektor Verlag 1999, ISBN: 3895760889

### **Links**

- **[www.digitalsimulator.de](http://www.digitalsimulator.de)**  
Es lassen sich einzelne Digitalschaltungen am Rechner aufbauen und ihre Funktion in einem Simulationslauf verfolgen (auch für das Wahlgebiet "Technische Informatik" geeignet). Eine umfangreiche Materialsammlung und eine weitere Linkseite führen weiter.
- **<http://www.a-rostin.de/klogic/indexd.html>**  
Mit dem Programm klogic lassen sich einzelne Digitalschaltungen am Rechner aufbauen und ihre Funktion in einem Simulationslauf verfolgen (auch für das Wahlgebiet "Technische Informatik" geeignet).
- **<http://www.oberstufeninformatik.de/dc/>**  
Der "didaktische Mikroprozessor" ist ein Simulationsprogramm, mit dem man den grundsätzlichen Aufbau eines Prozessors kennen lernen kann und in die entsprechende Assemblerprogrammierung eingeführt wird.
- **<http://www.kle.nw.schule.de/gymgoch/faecher/informat/index.html>**  
Hier werden die einzelnen Komponenten eines Mikroprozessorsystems vorgestellt. Zusätzlich steht ein Simulationsprogramm zur Verfügung.
- **<http://www.z80.info/z80emu.htm>**  
Eine Fülle von Z80 – Emulatoren und Simulationen von Geräten, die mit einer Z80-CPU arbeiteten (z.B. ZX80), stehen hier für alle Betriebssysteme zur Verfügung.
- **[http://www.informatik.hu-berlin.de/Institut/struktur/signalverarbeitung/lehre/ti1/ti1\\_d/prak/mp/mp-sim/mp-sim\\_vor.pdf](http://www.informatik.hu-berlin.de/Institut/struktur/signalverarbeitung/lehre/ti1/ti1_d/prak/mp/mp-sim/mp-sim_vor.pdf)**  
Humboldt-Universität zu Berlin, Praktikum Technische Informatik I Versuch MP-SIM Ausgabe 20.12.04 Mikroprozessor - Simulation Anleitung zur Versuchsvorbereitung und Versuchsdurchführung
- **<http://www.b.shuttle.de/b/humboldt-os/kokavec/eurocom>**  
Motorola 6802 - Mikroprozessor-Simulationsprogramm im Textmodus (DOS) mit unabhängig arbeitendem Crossassembler. Es wird das Mikroprozessor-Minimalsystem "Eurocom I" der Firma Eltec simuliert (ca. 1982), das an der Humboldtschule noch als Realsystem zur Verfügung steht. Zur Zeit entsteht ein neues System in Java (Ziel: Herbst 2005).



## L Informatik und Gesellschaft

Informations- und Kommunikationstechnologien (IKT) wirken in allen Lebensbereichen. Die Informatik kann und darf sich nicht in den Turm einer technischen Ingenieurlehre zurückziehen und die Folgen ihres Tuns von Anderen begutachten lassen. Sie muss die Verantwortung für diese Folgen zumindest teilweise mittragen. Die Schulinformatik sollte ihr darin folgen und der Herausforderung begegnen, neben langfristiger Berufsvorbereitung und der wichtigen Förderung formalistischer Denkwerkzeuge über Probleme der Lebenswelt von Schülern aufzuklären, die sich aus unbedarftem Umgang mit Informationstechnologien ergeben können. Nur so kann Schule die Schülerinnen und Schüler aus ihrer nicht ganz selbstverschuldeten Unmündigkeit heraus begleiten.

Informatik und Gesellschaft (I&G)		
Fachinhalte	Kompetenzen	Vernetzungen
<p>Zu den Themen im Bereich Informatik und Gesellschaft gehören:</p> <ul style="list-style-type: none"> <li>• Geschichte</li> <li>• Ethik</li> <li>• Informationsgesellschaft</li> <li>• Informationsrecht</li> <li>• Computerkunst</li> <li>• Ökologie</li> </ul>	<p>Die Kompetenzen sind in verschiedene Kompetenzstufen gestaffelt, die aufeinander aufbauen.</p> <ul style="list-style-type: none"> <li>• Die wichtigsten Diskurse im Bereich I&amp;G kennen und benennen.</li> <li>• Die zentralen Probleme identifizieren, die in diesen Diskursen angesprochen werden.</li> <li>• Die Grundbegriffe verstehen, mit denen diese Probleme in verschiedenen Quellen beschrieben werden.</li> <li>• Dieses Fachvokabular in Diskussionen und Referaten aktiv verwenden.</li> <li>• Beziehungen zu konkreten Fällen der Lebenswelt herstellen und mit dem spezifischen Vokabular beschreiben.</li> <li>• Die technischen Hintergründe kennen.</li> <li>• Aufbau und Funktionsweise dieser Technik verstehen.</li> <li>• Die Technik im Rahmen schulischer Möglichkeiten erstellen und modifizieren.</li> <li>• Zusammenhänge von IKT und Gesellschaft verstehen.</li> <li>• Einflussmöglichkeiten und Grenzen technischer Gestaltung erkennen.</li> </ul>	<ul style="list-style-type: none"> <li>• Themen aus dem Bereich Informatik und Gesellschaft lassen sich mit jedem anderen informatischen Thema kombinieren. Datenbanken passen zum Datenschutzrecht, Multimedia zum Urheberrecht, Hardware zu Ökologie, Software-Entwicklung zu Ethik und der Verantwortung, stabile Systeme zu konstruieren usw.</li> <li>• Als Orientierung kann bei einem gegebenen Thema die Frage dienen, welche gesellschaftlichen Wechselwirkungen in der Fach- und Tagespresse diskutiert werden.</li> <li>• Hintergrundinformationen und Lehrmaterialien sind im Anschluss z.B. über den <i>Deutschen Bildungsserver</i> zu beziehen.</li> <li>• I&amp;G ist von sich aus fachübergreifend. Eine Zusammenarbeit mit anderen Unterrichtsfächern – nicht nur mit der Mathematik – ist bei I&amp;G-Themen einfach zu realisieren, z.B. im Rahmen von fächerübergreifenden Projekten.</li> </ul>

## Kommentierte Bücher- und Linkliste

### Bücher

#### Geschichte

- **Abbate, Janet:** *Inventing the Internet*. Cambridge/Mass.: MIT Press, 1999. Die Geschichte des Internets von den Wurzeln bis zum WWW.
- **Ceruzzi, Paul E.:** *A History of Modern Computing*. Cambridge/Mass.: MIT Press, 1998. Das Standardwerk zur Geschichte des digitalen Rechners.
- **Hafner, Katie; Lyon, Matthew:** *Arpa Kadabra. Die Geschichte des Internet*. Heidelberg: dpunkt, 1997 (1996). Die Geschichte des ARPANET, von den ersten Knoten bis zum TCP/IP-Protokoll.

#### Ethik

- **Bowyer, Kevin W.:** *Ethics and Computer. Living Responsibly in a Computerized World*. IEEE Computer Society Press 1996. Materialreiche Diskussion ethischer Fragen beim Computereinsatz.
- **Weizenbaum, Joseph:** *Die Macht der Computer und die Ohnmacht der Vernunft*. Frankfurt/Main: Suhrkamp, 1978 (1976). Der Klassiker zum Thema Informatik, Computer und Verantwortung
- **Langford, Duncan:** *Practical Computer Ethics*. McGraw-Hill, 1995. Sehr praktisch orientierte Einführung in die wichtigsten ethischen Entscheidungsfelder.
- **Levy, Steven:** *Hackers. Heros of the Computer Revolution*. New York: Delta, 1984. Ein Buch aus einer Zeit als Hacker noch Künstler mit ethischen Grundsätzen waren und nicht jene Cybervandalen, die heute mit ihren Skripten die Netze verwüsten.

#### Informationsgesellschaft

- **Beck:** *IT-und Computerrecht (CompR)*. DTV-Beck, 2004. Die wichtigsten Gesetzestexte rund um IKT, praktisch der Quelltext der Informationsgesellschaft.
- **Kleinwächter, Wolfgang:** *Macht und Geld im Cyberspace*. Hannover: Heise 2004. Die Dokumentation des World Summit on Information Society (WSIS), auf dem die Schlüsselthemen der zukünftigen Informationsgesellschaft verhandelt werden.
- **Klumpp, Dieter; Kubicek, Herbert K; Roßnagel, Alexander (Hg.):** *next generation information society? Notwendigkeit einer Neuorientierung*. Mössingen: Talheimer, 2003. In über 30 Beiträgen werden die aktuellen Diskussionen über Wege und Ziele der Informationsgesellschaft zusammen gefasst.
- **Negroponte, Nicholas:** *Total digital*. München: Bertelsmann, 1995. Negropontes Unterscheidung zwischen Bits und Atomen prägt bis heute die Diskussion um die Ökonomie der Informationsgesellschaft.
- **Schulzki-Haddouti, Chrisitane:** *Bürgerrechte im Netz*. Bonn: Bundeszentrale für politische Bildung, 2003. In den Beiträgen wird die Entwicklung der Bürgerrechte in der Informationsgesellschaft vorgestellt.

#### Datenschutz

- **Garfinkel, Howard:** *Database Nation. The Death of Privacy in the 21th Century*. Sebastopol: O'Reilly, 2000. Garfinkel zeigt die Gefahren, die entstehen, wenn immer mehr Datenbruchstücke in verschiedenen Datenbanken landen.
- **Schulzki-Haddouti, Christiane (Hg.):** *Vom Ende der Anonymität. Die Globalisierung der Überwachung*. Hannover: Heise, 2000.
- **Whitaker, Reg:** *Das Ende der Privatheit. Überwachung, Macht und soziale Kontrolle im Informationszeitalter*. München: Kunstmann, 1999. Der Handel mit privaten Daten bedroht gleichzeitig die Sphäre des privaten Lebens.

#### Kunst

- **Whois/MARS (Hg.):** *Digitale Transformationen. Medienkunst als Schnittstelle von Kunst, Wissenschaft, Wirtschaft und Gesellschaft*. Heidelberg: Whois 2004. Ein Sammelband mit Beiträgen um die gesellschaftliche Bedeutung von e-culture.

- **Kaufhold, Enno; Jakob, Johanna:** *Natürlich Künstlich*. Jovis, 2001. In diesem reich bebilderten Ausstellungskatalog werden fünf der international bekanntesten Computerkünstler vorgestellt.

### Ökologie

- **Kuehr, Ruediger; Williams, Eric:** *Computer and Environment. Understanding and Managing Their Impacts*. Kluwer Academic Publishers 2003. Von der Ökobilanz von Computerhardware über internationale Abkommen bis zu Möglichkeiten des Recycling deckt dieses Buch alle wichtigen ab.

### Links

#### Geschichte der Rechentechnik

Ein Blick in die Geschichte zeigt, dass die Welt, wie wir sie heute vorfinden, auch ganz anders hätte aussehen können. Dass wir nicht in der besten aller, sondern nur in einer der möglichen Welten leben.

*An illustrated History of Computing*

**<http://www.computersciencelab.com/ComputerHistory/History.htm>**

Ungewöhnlich reich bebildeter Überblick der Rechnergeschichte vom Abakus zum PC.

*Computer History Online*

**<http://www.weller.to/com>**

Trotz des englischen Titels ein deutschsprachiges Angebot mit vielen Bildern zur Geschichte der Computer.

*Deutsches Museum*

**<http://www.deutsches-museum.de/ausstell/dauer/inform/infor.htm>**

Dauerausstellung Informatik. Als besonderes Angebot gibt es Quicktime-Panorama-Filme von historischen Großrechenanlagen.

#### Ethik

Die Informatik darf sich nicht als reine Technik-Wissenschaft verstehen und die Verantwortung für Anwendung ihrer Produkte anderen überlassen. Sie muss diese Verantwortung dafür zumindest teilweise selbst übernehmen.

*Ethische Leitlinien der GI*

**[http://www.gi-ev.de/verein/struktur/ethische\\_leitlinien.shtml](http://www.gi-ev.de/verein/struktur/ethische_leitlinien.shtml)**

Ausgearbeitet vom Arbeitskreis Verantwortung. Verabschiedet am 29. Januar 2004.

*Die Hacker-Ethik*

**<http://www.ccc.de/hackerethics?language=de>**

Ursprünglich formuliert von S. Levy, hier in der Fassung vom Chaos Computer Club.

*Ethik und Informatik*

**<http://www.capurro.de/antritt.htm>**

Antrittsvorlesung von Rafael Cappuro, gehalten am 2.5.1990.

#### Informationsgesellschaft

Inzwischen zeichnen sich die Konturen der seit Jahrzehnten angekündigten Informationsgesellschaft immer deutlicher ab. Nur in ihrem Rahmen können Informationstechnologien verstanden werden.

*eEurope2005*

**[http://europa.eu.int/information\\_society/eeurope/2002/news\\_library/documents/eeurope2005/eeurope2005\\_de.pdf](http://europa.eu.int/information_society/eeurope/2002/news_library/documents/eeurope2005/eeurope2005_de.pdf)**

Eine Informationsgesellschaft für Alle. Ein Aktionsplan der EU-Kommission.

*Bundesregierung*

**<http://www.bundesregierung.de/Themen-A-Z/-,460/Informationsgesellschaft.htm>**

Informationsseite der Bundesregierung.

Daneben gibt es Schwerpunkte im *Bundesministerium für Bildung und Forschung (BMBF)*

**<http://www.bmbf.de/de/398.php>**

und dem *Bundesministerium für Wirtschaft und Arbeit (BMWA)*

**<http://www.bmwa.bund.de/Navigation/Technologie-undEnergie/informationsgesellschaft.html>**

*Statistisches Bundesamt*

**[http://www.destatis.de/themen/d/thm\\_infogesell.php](http://www.destatis.de/themen/d/thm_infogesell.php)**

Aktuelle Zahlen vom Statistischen Bundesamt

*WSIS*

**<http://www.itu.int/wsis>**

Auf dem World Summit on Information Society wird die Struktur einer globalen Informationsgesellschaft ab 2015 verhandelt.

**Informationsrecht**

Rechtliche Regelungen vermitteln zwischen verschiedenen Interessengruppen. Zu den wichtigsten Gesetzen zählen Datenschutzrecht, Urheberrecht, Äußerungsdelikte, Markenrecht, Vertragsrecht.

*Lehrer-Online*

**<http://www.lehrer-online.de/dyn/16.htm>**

Hier werden die rechtlichen Probleme von Schulen detailliert und übersichtlich diskutiert. Sicherlich die erste Adresse für Lehrer.

*DDI der Humboldt-Universität zu Berlin*

**<http://ddi.informatik.hu-berlin.de/>**

Unterrichtsmaterial zum Thema Informationsrecht.

*Berliner Datenschutzbeauftragter*

**<http://www.datenschutz-berlin.de/home.htm>**

Die Website des Berliner Datenschutzbeauftragten ist ein guter Ausgangspunkt, um sich über Datenschutz zu informieren.

*iRight*

**<http://www.iright.de/>**

Ein Portal zum Thema Urheberrecht, zugeschnitten auf die Bedürfnisse von speziell jugendlichen Verbrauchern. Es wird Anfang 2005 Online gehen.

**Computerkunst**

Computer erweitern die Möglichkeit künstlerischer Ausdrucksmittel. Die Kunst reflektiert das Selbstverständnis, die Wünsche, Ängste und Träume einer Gesellschaft lange vor jeder theoretischen Reflexion.

*DAM*

**<http://www.dam.org/>**

Das Digital Art Museum bietet nicht nur Informationen zur Geschichte der Computerkunst sondern hat auch eine Galerie in der Tucholskystr. mit wechselnden Ausstellungen.

*Ars Electronica*

**<http://www.aec.at/en/index.asp>**

Der jährliche Preis für Medienkunst.

*Der Computer als Werkzeug in der Kunst und Kunstwissenschaft*

**<http://www.math-inf.uni-greifswald.de/mathematik+kunst/themen.html>**

Eine Ausstellung des Instituts für Mathematik und Informatik der Ernst-Moritz-Arndt-Universität Greifswald.

## **Ökologie**

Auch die digitale Welt braucht analoge Technik. Was aber geschieht mit all den ausgedienten Chips, Kondensatoren, Platinen, Batterien, Bildschirmen und Tastaturen?

*ECO-COM*

**<http://www.tu-berlin.de/zek/wb/pdf/ecohbd.pdf>**

Computerarbeitsplatz, Gesundheit und Ökologie. Erstellt an der TU-Berlin, 2001.

*Ökobilanz eines PC*

**<http://www.learn-line.nrw.de/angebote/ergonomie/medio/oekopc.htm>**

Studienarbeit über Be- und Entlastungen eines PCs.

*Exporting Harm*

**<http://www.ban.org/#HighTechTrash>**

Der erschreckende Report des Basel Action Networks über Elektroschrottsorgung in Südchina.

*Elektroschrottverordnung*

**<http://www.zvei.org/index.php?id=34>**

Regierungsentwurf vom 1. September 2004.

## **Impressum**

Senatsverwaltung für  
Bildung, Jugend und Sport

Landesinstitut für  
Schule und Medien  
(LISUM)  
Alt Friedrichsfelde 60  
10315 Berlin  
Tel.: 9021-0

Beratungsstelle für  
Informationstechnische Bildung und  
Computereinsatz in Schulen (BICS)  
am LISUM

Redaktion: Helmut Witten

Verantwortlich: Holger Thymian

Berlin, Februar 2005

Alle Rechte vorbehalten. Nachdruck oder andere Formen der  
Vervielfältigung, auch auszugsweise, nur mit schriftlicher  
Genehmigung. Frei für die Nutzung durch die Berliner Schulen.