

Inhalt

V	Vorwort	
VII	Inhalt	
X	Liste der Unterrichtsbeispiele	
1	Einleitung — 1	
2	Der didaktische Rahmen — 6	
2.1	Bildungsziele und Fachdidaktik — 6	
2.2	Die Definition von Allgemeinbildung — 9	
2.3	Konstruktivismus als Lerntheorie und Richtschnur im Unterricht — 14	
2.4	Fundamentale Ideen zur Strukturierung des Unterrichts und des Lernprozesses — 16	
2.5	Standards und Kompetenzmodelle als überprüfbare Aufgabenkataloge — 21	
2.6	Anforderungen an einen aktuellen Satz Fundamentaler Ideen — 22	
2.7	Fundamentale Ideen der Schulinformatik — 23	
2.7.1	Algorithmisierbarkeit — 27	
2.7.2	Modellierbarkeit — 29	
2.7.3	Kontextualisierbarkeit — 30	
2.7.4	Digitalisierbarkeit — 32	
2.7.5	Vernetzbarkeit — 34	
2.7.6	Realisierbarkeit — 35	
2.8	Der Filter der Allgemeinbildung — 37	
2.9	Kompetenzbereiche der Schulinformatik — 39	
2.10	Informatik als technisches Fach — 44	
2.11	Fundamentale Ideen als Grundlage für die Unterrichtsgestaltung — 48	
3	Informatik und Gesellschaft — 50	
3.1	Das Bild der Informatik — 50	
3.2	Gesellschaftliche Folgen von Informatiksystemen — 56	
3.3	Datenbanksysteme — 64	
3.3.1	SQL als Abfragesprache — 64	
3.3.2	Zur Entwicklung von Datenmodellen — 66	
3.3.3	Beispiele zum Unterricht mit Datenbanken — 69	
3.4	Netze — 73	
3.4.1	Netzwerke — 73	
3.4.2	Codes und Verschlüsselung — 74	
3.4.3	Technische Grundlagen von Netzwerken — 75	

3.4.4	Algorithmen auf Graphen —	81
3.4.5	Maschinenlernen und Neuronale Netze —	82
4	Informatische Grundlagen —	88
4.1	Systeme —	88
4.2	Modelle —	89
4.2.1	Modellierung —	89
4.2.2	Der Zweck der Modellbildung —	95
4.2.3	Modelltypen und informatische Begriffsbildung —	98
4.2.4	Modellierungsbeispiel: Die elektronische Patientenakte ePA —	99
4.3	Syntax —	104
4.3.1	Syntax in Programmiersprachen —	105
4.3.2	Syntax in Anwendungen —	107
4.3.3	Syntax eigener Programmiersprachen —	108
4.4	Codes —	109
4.5	Technische Informatik —	117
4.5.1	Wozu technische Informatik? —	118
4.5.2	Einfache Schaltnetze —	119
4.5.3	Realisierung von Mealyautomaten —	122
4.5.4	Programmierbare Rechenwerke —	128
4.6	Nebenläufige Prozesse —	132
4.6.1	Synchronisation mit gegenseitigem Ausschluss —	135
4.6.2	Synchronisation durch Kooperation —	137
4.6.3	Notation von nebenläufigen Prozessen —	139
4.7	Grenzen von Informatiksystemen —	144
4.7.1	Numerische Grenzen —	145
4.7.2	Zeitliche Grenzen —	145
4.7.3	Modellierungsgrenzen —	146
4.7.4	Ethische und moralische Grenzen —	148
4.7.5	Grenze der Algorithmisierbarkeit —	149
5	Programmieren —	153
5.1	Was bedeutet Programmieren? —	153
5.2	Die Automatisierung realer Miniwelten —	157
5.2.1	Technische Systeme —	158
5.2.2	Die Rekonstruktion technischer Systeme —	159
5.3	Fortgesetzter Programmierunterricht —	165
5.3.1	Beobachtungen aus der Praxis —	165
5.3.2	Der Übergang zum selbstständigen Programmieren —	169
5.3.3	Aufgaben für Fortgeschrittene und Bottom-Up- / Top-Down-Entwicklung —	172

5.3.4	Programmieren als persönliche Ausdrucksmöglichkeit —	174
5.4	Programmierparadigmen —	176
5.5	Weshalb Objektorientierte Programmierung? —	177
5.5.1	OOP und schulische Anforderungen —	177
5.5.2	Klassen- vs. Prototypen-Modell —	179
5.6	Visuelle Programmiersprachen —	184
5.7	Beispiele —	186
6	Methodik —	191
6.1	Fachbezogene Methodik —	191
6.2	Informatik im Kontext —	192
6.3	Offene Aufgaben und Selbstdifferenzierung —	201
6.4	Bewertung von offenen Aufgaben —	203
6.5	Öffnung von Aufgaben —	205
6.5.1	Öffnung von Aufgaben mit präzise definiertem Zielprodukt —	206
6.5.2	Öffnung des Unterrichts durch Unterrichtsprojekte —	208
6.5.3	Ergebnisse offener Aufgaben teilen —	210
6.6	Produktorientierung und Realisierung —	211
6.7	Teamarbeit —	215
6.8	Zwei ausführliche Beispiele —	219
6.8.1	Beispiel: „Alarm auf dem Bauernhof“ —	220
6.8.2	Beispiel: „Bildkompression“ —	223
6.9	Konzepte erfahrungsbezogen lehren —	224
6.10	Instruktion – wenn, dann schüleraktivierend —	227

Literaturverzeichnis —	231
-------------------------------	------------

Index —	235
----------------	------------