

# ABZ-Lehrplan für das Ergänzungsfach Informatik



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

© 2008 ABZ  
19. Februar 2009

## **Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich**

Der ABZ-Lehrplan für das Ergänzungsfach Informatik dient – ähnlich wie die kantonalen Lehrpläne – der Umsetzung des Rahmenlehrplans.

Das Ausbildungs- und Beratungszentrum für Informatikunterricht (ABZ) der ETH Zürich stellt diesen Lehrplan für das Ergänzungsfach Informatik den schweizerischen Gymnasien kostenlos als Vorlage für die Erstellung eigener Lehrpläne zur Verfügung. Die Schulen dürfen den Wortlaut gekürzt oder ungekürzt übernehmen, sofern sie das ABZ als Quelle nennen und dem ABZ unaufgefordert ein Exemplar zusenden.

Nähere Angaben zum Ausbildungs- und Beratungszentrum für Informatikunterricht finden Sie auf der Webseite: [www.abz.inf.ethz.ch](http://www.abz.inf.ethz.ch).

## 1 Vorbemerkungen

Der ABZ-Lehrplan für das Ergänzungsfach Informatik verabschiedet sich endgültig von der Vermittlung von ICT-Kompetenzen und von kurzlebigen Produktwissen. In Bezug auf Unterrichtskonzepte und Ansprüche schließt sich der Informatikunterricht der Mathematik und den naturwissenschaftlichen Fächern an. Die Mathematik und die Naturwissenschaften bauen ihren Unterricht nicht anhand der letzten Neuheiten und der kommerziellen Anwendungen auf, sondern in sorgfältiger Begriffsbildung und in der Vermittlung grundlegender Konzepte, Methoden und Erkenntnisse. Nur auf diese Weise können ein langlebiges Wissen und ein tiefes Verständnis der Grundlagen erreicht werden. Genau diesen Weg verfolgen wir mit dem beiliegenden Lehrplan für das Ergänzungsfach Informatik. Zusätzlich nutzen wir den Informatikunterricht, um überschaubare Brücken zwischen der mathematisch-naturwissenschaftlichen Denkweise und der Vorgehensweise der Ingenieurwissenschaften zu schlagen.

Weil die Vorkenntnisse der Lehrpersonen sehr unterschiedlich sind und die Fachleute sich auf Grund der dynamischen Entwicklung unterschiedlicher Informatikbereiche nicht auf eine kleine Menge von Themenbereichen einigen können, besteht unser Lehrplan aus einem Kernteil und aus mehreren zur Auswahl stehenden Vertiefungsmodulen. Der Kernteil ist Pflicht und hat den Umfang von ungefähr zwei Jahreswochenstunden (2 Semester zu je 2 Wochenstunden). Die Module, die individuell zur Auswahl stehen, sind unter den Vertiefungsthemen aufgeführt. Einige dieser Vertiefungsthemen beinhalten mehrere Unterrichtsmodule, andere entsprechen nur einem einzigen Modul. Die Unterrichtsmaterialien, die vom ABZ entwickelt worden sind, ermöglichen eine individuelle Auswahl der Vertiefungsstufen. Abhängig vom Vertiefungsgrad können in einem Schuljahr 4–8 Module behandelt werden. Das ABZ bietet drei fertige Module an und entwickelt und testet laufend weitere der hier vorgestellten Module.

## 2 Allgemeines

Die Informatik als Grundlagenwissenschaft untersucht die Grenzen der Automatisierbarkeit und die quantitativen Gesetze der Informationsverarbeitung. Sie entwickelt algorithmische Methoden zur Lösung von Problemen sowie Konzepte zur Datenverwaltung und sicheren Datenkommunikation. Sie setzt diese Methoden und Konzepte mittels Programmieren um. Als angewandte Ingenieurdisziplin durchdringt sie mit ihren Softwaresystemen alle Bereiche des Lebens.

Das Ziel des Informatikunterrichts liegt in der Verbindung der mathematisch-naturwissenschaftlichen Denkweise mit dem pragmatischen, zielorientierten

Vorgehen der Ingenieurwissenschaften in einem Fach. Das Ergänzungsfach Informatik vermittelt Kompetenzen in folgenden Bereichen: eindeutige, formale Beschreibung der Problemstellungen, Abstraktion und Modellierung, Suche nach algorithmischen Lösungswegen, Umsetzung von Lösungsmethoden mittels Programmieren, Messen der Effizienz der entwickelten Algorithmen, Verifikation und Testen der entworfenen Systeme. Im Vordergrund stehen die algorithmische Denkweise sowie die Beherrschung der modularen Entwurfstechnik für komplexe Systeme, die ein Grundkonzept der technischen Disziplinen darstellt.

Im Ergänzungsfach Informatik werden die Grundlagen des Programmierens, der Algorithmik, der Informationssysteme und der Datenverwaltung sowie der sicheren Kommunikation und der Rechnerarchitektur vermittelt.

### **Stundendotation**

Der Lehrplan beinhaltet Lerninhalte für das Ergänzungsfach Informatik und soll in den folgenden Schulstufen eingesetzt werden:

- Kurzzeitgymnasium: 3. und 4. Klasse, jeweils 2–3 Stunden pro Woche (1. und 2. Semester),
- Langzeitgymnasium: 5. und 6. Klasse, jeweils 2–3 Stunden pro Woche (1. und 2. Semester).

## **3 Bildungsziele**

Der Unterricht in Informatik

- erweitert unsere Weltanschauung durch einen neuen Blickwinkel der Automatisierung und deren Grenzen;
- fördert die Erkenntnis, dass vielfältige Probleme aus dem Alltag, der Wirtschaft, der Gesellschaft, der Technik und der Wissenschaften Probleme der Informationsverarbeitung, der Informationsdarstellung und der Informationsübertragung sind. Zur Lösung dieser Probleme werden Methoden (Algorithmen) entwickelt und als Programme implementiert. Die Programme werden anschliessend auf ihre Korrektheit überprüft;
- schult das algorithmische Denken als die Grundlage der Automatisierung;
- zeigt, wie die Qualität der algorithmischen Lösungswege anhand der Berechnungskomplexität von Programmen gemessen wird. Weiter sollen Programme als Softwareprodukte nach unterschiedlichen Kriterien bewertet und weiterentwickelt (verbessert) werden;

- lehrt, Lösungsmethoden (Algorithmen) in eine Programmiersprache umzusetzen. Das Programmieren wird als Kommunikation mit dem Rechner und damit als Steuerung von Maschinen verstanden. Weil die Maschine keine Improvisationsfähigkeit besitzt, erlernen die Schülerinnen und Schüler dabei ein Vorgehen eindeutig zu beschreiben;
- schult die modulare Vorgehensweise beim Entwurf von komplexen Systemen und fördert strukturiertes Denken, was massgebend für technische Disziplinen ist;
- zeigt die Wichtigkeit und die Wege der Verifikation (des Testens) von entwickelten Systemen;
- verbindet die exakte analytische Denkweise der Mathematik mit dem algorithmischen Denken und dem zielgerichteten Vorgehen der Ingenieurwissenschaften;
- bildet eine Grundlage für das Verständnis der hoch technisierten Welt.

## 4 Richtziele

### Grundkenntnisse

Die Maturandinnen und Maturanden kennen

- die Grundbegriffe und Grundkonzepte der Informatik wie Algorithmus, Programm, von-Neumann-Rechner, Compiler sowie die Berechnungskomplexität und wissen, dass die Suche nach Lösungen nicht immer automatisierbar ist,
- die Geschichte der Informatik und die Motivation zur Entwicklung grundlegender Informatikkonzepte,
- grundlegende Methoden zum Entwurf von Algorithmen für die Lösung von unterschiedlichen Problemen,
- grundlegende Programmierkonzepte, die sie in einer konkreten Programmiersprache umsetzen können,
- verschiedene Datenstrukturen und ihre Einsatzmöglichkeiten für eine effiziente Implementierung von Algorithmen,
- unterschiedliche Such- und Sortieralgorithmen sowie deren Programmierung,
- Methoden zur Datenstrukturierung und für die Datenkommunikation, dynamische Datenstrukturen zur Datenverwaltung und das Grundkonzept von relationalen Datenbanken,

- die Bedeutung des Datenschutzes und der Sicherheit in der vernetzten Welt, die Grundkonzepte der Kryptographie und wissen, wie sichere Kommunikation über öffentliche Internetverbindungen realisiert werden kann,
- das Konzept der Objektorientierung zum modularen Entwurf von Softwaresystemen.

### **Grundfertigkeiten**

Die Maturandinnen und Maturanden können

- Probleme der Informationsverarbeitung in verschiedenen Bereichen erkennen, darstellen, analysieren und algorithmisch lösen,
- Algorithmen in einer höheren Programmiersprache implementieren und anschließend testen,
- die grundlegenden Datentypen zur Implementierung von Algorithmen nach sorgfältigem Vergleich bezüglich Effizienz auswählen und auswerten,
- dynamische Datenstrukturen wie Listen und Bäume zur Datenverwaltung auswählen und einsetzen sowie eine relationale Datenbank aufbauen,
- einfache Kryptosysteme nachvollziehen und nach ihrem Sicherheitsgrad bewerten,
- in einer Gruppenarbeit die Planung, die Analyse, die Implementierung und das Testen eines Informatikprojektes durchführen und das Vorgehen dokumentieren.

### **Grundhaltungen**

Die Maturandinnen und Maturanden sind sich bewusst, dass

- nicht alle Aufgaben der Informationsverarbeitung mit Hilfe von Algorithmen lösbar sind und sollten diese Tatsache immer in Betracht ziehen,
- die algorithmische Denkweise die Basis ist für die erfolgreiche Automatisierung von Prozessen aller Art,
- systematisches und modulares Vorgehen massgebend für den Entwurf von Softwaresystemen und Algorithmen ist,
- eine sorgfältige Dokumentation des eigenen Vorgehens wesentlich ist für die zukünftige Erweiterung, Änderung und Verbesserung der eigenen Produkte,
- Datenschutz und sichere Kommunikation unbedingt angestrebt werden müssen,

- die Mathematik ihnen viele wichtige Werkzeuge für das Entwerfen und Entwickeln von zuverlässiger und effizienter Software liefert.

Die Maturandinnen und Maturanden sind bereit,

- eine Programmiersprache zu verwenden, um die Arbeit des Rechners zu steuern,
- sich den Schwierigkeiten und Anforderungen angewandter Probleme zu stellen und die ausgearbeiteten Lösungen ständig zu verbessern,
- sich mit den Auswirkungen der Informatik im Alltag auseinanderzusetzen,
- unterschiedliche Lösungsansätze auszuprobieren und nach effizienten Lösungsmethoden zu suchen,
- die Effizienz von vorgeschlagenen Lösungswegen zu untersuchen,
- die Produkte der eigenen Arbeit kritisch zu beurteilen und sich mit den Möglichkeiten der Weiterentwicklung und Verbesserung zu beschäftigen.

## 5 Lerninhalte

Unser Lehrplan besteht aus einem Kernteil, der Pflicht ist, und aus mehreren zur Auswahl stehenden Vertiefungsmodulen.

### 5.1 Kernthemen

Die folgenden Kernthemen sollen mindestens im skizzierten Umfang unterrichtet werden.

	Lerninhalte	Querverweise
<b>Programmierung I</b>	<p>Einführung in die strukturierte Programmierung im Kleinen. Umsetzung in einer Programmiersprache (vorzugsweise Pascal/Delphi).</p> <ul style="list-style-type: none"> <li>– Grundbegriffe und Konzepte: Programm, Rechnerbefehle, Unterprogramm, Schleifen (Iteration), Parameter, Variable, bedingte Verzweigungen, bedingte Schleifen, Rekursion.</li> </ul>	<p>Mathematik (Logik, Iteration, Rekursion)</p>
<b>Geschichte der Informatik</b>	<p>Die Geschichte der Informatik und der fundamentalen Konzepte.</p> <ul style="list-style-type: none"> <li>– Bildung der Grundbegriffe: Algorithmus und Programm (Turing, Church), Unvollständigkeitssatz der Mathematik (Gödel), Modell des Computers (von Neumann), Sprache des Computers (Assembler), Unentscheidbarkeit (algorithmische Unlösbarkeit von Problemen), Berechnungskomplexität, Beschreibungskomplexität, korrekte Argumentation und Logik (direkte und indirekte Beweise).</li> </ul>	<p>Geschichte (Hilbert, Gödel, Turing, von Neumann)</p> <p>Mathematik (Logik)</p> <p>Physik (Elektromagnetismus)</p>
<b>Datenstrukturen und Algorithmen</b>	<p>Einführung in die grundlegenden Datenstrukturen zur effizienten Implementierung von Algorithmen.</p> <ul style="list-style-type: none"> <li>– Darstellung von Daten im Computer, Organisation des Speichers.</li> <li>– Fundamentale Datenstrukturen (Felder, Listen, Bäume, Graphen, Mengen), Verwendung von Zeigern (Pointer).</li> <li>– Wahl der geeigneten Datenstrukturen zur Implementierung von Algorithmen für einfache Aufgaben der Datenverarbeitung im Hinblick auf die Effizienz des resultierenden Programms.</li> <li>– Grundlegende Algorithmen wie zum Beispiel Such- und Sortieralgorithmen.</li> </ul>	<p>Mathematik (Kombinatorik, Analysis, Reihen und Summen, exponentielle und logarithmische Funktionen)</p>

## 5.2 Vertiefungsthemen

Die folgenden Vertiefungsthemen stehen zur Auswahl. Die ersten drei (Programmierung II, Informationssysteme und Internet sowie Algorithmik) bestehen aus mehreren unabhängigen Modulen, von denen wiederum einige ausgesucht werden können.

Je nach Vertiefungsgrad sollen 4–8 Vertiefungsmodule in einem Schuljahr behandelt werden.



	Lerninhalte	Querverweise
<b>Programmierung II</b>		
<ul style="list-style-type: none"> <li>• Einführung in das objektorientierte Programmieren</li> </ul>	Objektorientierung und Modularität bei der Systementwicklung im Software-Engineering. <ul style="list-style-type: none"> <li>– Begriffe Objekt und Klasse.</li> <li>– Durchführung eines kleinen Softwareprojektes (Gruppenarbeit, Planung, Schnittstellen, Implementierung, Dokumentation, Testen).</li> </ul>	Mathematik (Mengenlehre)
<ul style="list-style-type: none"> <li>• Programmieren von mathematischen Methoden</li> </ul>	Implementierung von numerischen Methoden. <ul style="list-style-type: none"> <li>– Numerische Algorithmen (Suchen von Nullstellen, Schnittpunkte von Kurven, Berechnung von Polynomen, Interpolation, Integralrechnung).</li> <li>– Methoden der linearen Algebra (Systeme von linearen Gleichungen und Ungleichungen, diskrete Optimierung, Vektor- und Matrixmultiplikation).</li> <li>– Methoden der analytischen Geometrie.</li> </ul>	Mathematik (Numerik, lineare Algebra, Analysis, analytische Geometrie)
<ul style="list-style-type: none"> <li>• Programmieren von Robotern</li> </ul>	Implementierung von Steuerungen. <ul style="list-style-type: none"> <li>– Hardware/Software-Schnittstelle, Programmieren von Bewegungen, Entwicklung von Verhaltensstrategien, Spieltheorie.</li> </ul>	Physik (Mechanik)
<b>Informationssysteme und Internet</b>		
<ul style="list-style-type: none"> <li>• Datenstrukturen zur dynamischen Datenverwaltung</li> </ul>	Einführung in die grundlegenden dynamischen Datenstrukturen für die Datenverwaltung. <ul style="list-style-type: none"> <li>– Grundoperationen der Suche, des Einfügens und der Entfernung von Elementen und deren effiziente Umsetzung abhängig von der Datenorganisation (balancierte Bäume, Hashing).</li> </ul>	Mathematik (Kombinatorik, logarithmische Funktionen)
<ul style="list-style-type: none"> <li>• Relationale Datenbanken</li> </ul>	Einführung in relationale Datenbanken. <ul style="list-style-type: none"> <li>– Der Begriff der Relation, das Konzept der relationalen Datenbanken (Attribute, Werte).</li> <li>– Ein kleines Projekt zum Entwurf einer Datenbank.</li> </ul>	Mathematik Wirtschaft

	<b>Lerninhalte</b>	<b>Querverweise</b>
• Suchalgorithmen im Internet	Einführung über die Suche im Internet und die Bewertung der Suchresultate. – Grundlegende Suchstrategien. – Statistische Methoden für Seitenrankings.	Mathematik (Statistik, Kombinatorik)  Wirtschaft
<b>Algorithmik</b>		
• Entwurf von Algorithmen und Analyse der Effizienz	Einführung in die grundlegenden Methoden zum Entwurf von Algorithmen mit dem Fokus auf die Berechnungskomplexität (Effizienz) und ihre Analyse. Grundmethoden:  – Greedy als Optimierungsmethode, – Teile und Herrsche als rekursive Methode, – Dynamisches Programmieren als iterative Methode, – Lokale Suche in der Optimierung, – Backtracking als vollständige Suche.	Mathematik (Kombinatorik, rekursive Funktionen, Reihen, diskrete Optimierung)  Wirtschaft (Optimierung)
• Geometrische Algorithmen	Einführung in die geometrischen Algorithmen für einige grundlegende geometrische Probleme. – Suche nach zwei Punkten mit minimaler Entfernung, – Konvexe Hülle, – Triangulierung eines planaren Objektes, – Kunstgalerie-Problem, – Postamt-Problem.	Mathematik (Geometrie)  Geografie  Wirtschaft (Optimierung)
• Zufallsgesteuerte Algorithmen	Einführung in die zufallsgesteuerten Algorithmen. – Zufall als Mittel zur Beschleunigung der Berechnungen. – Begriffe der Fehlerwahrscheinlichkeit und der erwarteten Komplexität.	Mathematik (Wahrscheinlichkeitsrechnung und Statistik)
<b>Aufbau und Funktionsweise eines Computers</b>	Modell des Computers (von-Neumann-Architektur) und Entwurf von elektronischen Schaltkreisen. – Organisation des Rechners (Prozessor, Speicher, Busse, Eingabe, Ausgabe). – Organisation des Speichers (Register, Speicherung von Zahlen und Symbolen). – Rechnerarithmetik. – Struktur des Prozessors. – Darstellungen von booleschen Funktionen. – Modularer Entwurf von elektronischen Schaltkreisen (Multiplexer, Halbaddierer, Addierer).	Mathematik (Boolesche Algebra, Logik, Zahlensysteme, Arithmetik)  Physik (Elektrizität)

	<b>Lerninhalte</b>	<b>Querverweise</b>
<b>Kryptologie</b>	<p>Einführung in die sichere Kommunikation und den Aufbau von Kryptosystemen.</p> <ul style="list-style-type: none"> <li>– Vermittlung von Grundbegriffen: Sender, Empfänger, Klartext, Kryptotext, Verschlüsselung, Entschlüsselung, Schlüssel.</li> <li>– Kurze Geschichte der Geheimschriften (Caesar, Richelieu) und Kerkhoffsches Prinzip.</li> <li>– Symmetrische Kryptosysteme, ihre Stärken und Schwächen, Attacken gegen monoalphabetische Systeme, One-Time-Pad.</li> <li>– Der Begriff eines sicheren Kryptosystems anhand der Algorithmik und des Konzeptes der Berechnungskomplexität.</li> <li>– Schlüsselaustausch über unsichere Kanäle (Diffie-Hellman).</li> <li>– Public-Key-Kryptographie und der Begriff der Einweg-Funktionen. Digitale Unterschriften und andere Anwendungen im E-Commerce.</li> </ul>	<p>Mathematik (Algebra, Zahlentheorie, Statistik, Kombinatorik)</p> <p>Geschichte</p> <p>Wirtschaft (E-Commerce)</p>
<b>Automatenentwurf</b>	<p>Einführung in den Entwurf und die Verifikation von endlichen Automaten.</p> <ul style="list-style-type: none"> <li>– Vermittlung der Grundbegriffe: Alphabet, Symbole, Wörter, Sprachen, Syntax, Semantik, Relation, Entscheidungsprobleme.</li> <li>– Das Konzept des endlichen Automaten als einfaches Rechnermodell. Begriffe: Konfiguration, Berechnungsschritt und Berechnung.</li> <li>– Entwurfsmethoden für Automaten mit dem Fokus auf den modularen Entwurf.</li> <li>– Verifikation der Korrektheit mittels Induktionsbeweisen.</li> <li>– Ein kleines Projekt: Entwurf eines Automaten für die Steuerung von Liften und Ampelkreuzungen.</li> <li>– Beweise der Nichtexistenz von Automaten für gewisse Aufgabenstellungen.</li> </ul>	<p>Mathematik (Logik, Kombinatorik, Induktion)</p> <p>Linguistik</p>

	<b>Lerninhalte</b>	<b>Querverweise</b>
<b>Berechenbarkeit</b>	<p>Einführung in die Berechenbarkeit mit dem Ziel, die Grenzen der Automatisierbarkeit aufzuzeigen.</p> <ul style="list-style-type: none"> <li>– Geschichte der Suche nach algorithmisch unlösbaren Aufgaben.</li> <li>– Das Konzept der Unendlichkeit, unterschiedliche unendliche Grössen, Diagonalisierungsmethode.</li> <li>– Methode der Reduktion.</li> <li>– Algorithmische Unlösbarkeit des Testens von Programmen.</li> </ul>	<p>Mathematik (Logik)</p> <p>Geschichte</p>
<b>Computergrafik</b>	<p>Einführung in die Computergrafik</p> <ul style="list-style-type: none"> <li>– Koordinatensystem und affine Transformationen.</li> <li>– Darstellung der geometrischen Objekte (Linien, Kreise, Ellipsen, Bézierkurven).</li> <li>– Raster- und Vektorgrafik.</li> <li>– Farbmodelle (RGB, CMYK).</li> </ul>	<p>Mathematik (Vektorgeometrie, Geometrie)</p>
<b>Computernetze</b>	<p>Einführung in die Problematik der Kommunikation in Rechnernetzen.</p> <ul style="list-style-type: none"> <li>– Kodierung der Nachrichten, selbstverifizierende Codes (Hamming), Signaltheorie.</li> <li>– OSI-Schichtenmodell.</li> <li>– Entwurf und Charakteristiken von Netzen (Bandbreite, Topologie).</li> <li>– Kommunikationsverfahren (Routing, Wahlverfahren, Gossiping).</li> </ul>	<p>Mathematik</p> <p>Physik (Elektrotechnik)</p>