

Informatik

Was eigentlich ist Informatik?

Statt einer längeren Antwort ([s.u.](#)) auf diese eine kurze Antwort auf eine andere Frage:

Wo im Alltag steckt überall Informatik drin? Hier eine kleine Auswahl:

- Google & Co.: Sortier- und Suchalgorithmen, Datenbanken, ...
- WWW, E-Mail, Chat & Co: Server-Client-Architektur, Datentransport in Netzen, ...
- Routenplaner: Problem des kürzesten oder schnellsten Weges
- Bankgeschäfte: Datenbanken, Sichere Datenübertragung, Verschlüsselung, ...
- Amazon, eBay & Co: Dynamisches HTML, Datenbanken, Verschlüsselung,
- CD-Player: Fehlererkennende oder -korrigierende Codes
- MP3, Zip: Datenkompression, verlustfrei oder verlustbehaftet

Inhalt

- [Einführungsphase](#)
- [Qualifikationsphase](#)

Einführungsphase

Zu Beginn eine extrem wichtige Klärung:

Informatikunterricht ist nicht:

- Schulung in der Bedienung eines Programms (Textverarbeitung, Tabellenkalkulation, Präsentation, Bildbearbeitung, usw.)
- Surfen oder Suchen im Internet,
- Beratung bei Computer- oder Softwareproblemen,

Informatikunterricht ist weniger praktisch, mehr theoretisch ausgerichtet. Manchmal findet der Unterricht ganz ohne Rechner statt, nur mit Stift und Papier oder Kreide und Tafel. Von [E.W. Dijkstra](#), einem niederländischen Informatiker, stammt die klassische Begründung dafür:

"Computer science is no more about computers than astronomy is about telescopes." (In der Informatik geht es genau so wenig um Computer, wie in der Astronomie um Teleskope.)

Worum geht es dann? Das erfahren Sie im Folgenden.

Die Themen der beiden Semester der E-Phase sind

- HTML, Skriptsprachen, Modellrechner (E1)
- Algorithmen I und Sequentielle Datenstrukturen (E2)

Informatik

HTML, Skriptsprachen, Modellrechner (E1)

1. [HTML](#) ist die Sprache des WWW, eines Teils des Internets. **HTML** ist eine Abkürzung für *HyperText Markup Language*, zu deutsch "Auszeichnungssprache für Über-Text". (Damit ist gemeint, dass diese Art von Text *über* den üblichen – linearen – Text von Papiermedien hinausgeht, indem er etwa Texte durch Links verknüpft zu einem *Informations-Netz*). HTML steckt in jeder Webseite, die man mit einem Web-Browser (Internet-Explorer, Firefox, Opera) aufruft. In den Webseiten werden einzelne Textteile (z.B. Überschriften, Absätze, Tabellen, Links) durch *Marken* ausgezeichnet.

Hier lernen Sie die Grammatik dieser Auszeichnungssprache kennen und wie man damit Webseiten gestaltet. Das **Hauptziel** besteht jedoch darin, Sie mit der *abstrakt-formalen Denk- und Arbeitsweise der Informatik* bekannt und vertraut zu machen. Es wird ein einfacher Editor (Notepad, [SciTE](#)) benutzt, kein Profiwerkzeug.

2. [Skriptsprachen](#). Es gibt zwei Arten von Webseiten: Statische und dynamische, also starre, unveränderliche (z.B. die Homepage unserer Schule, ausgenommen die Menüs) und veränderbare (z.B. die Seiten einer Online-Firma, die Produkte verkauft). Die Veränderungen können in Kleinigkeiten bestehen, z.B. einem unterschiedlichen Begrüßungstext je nach Tageszeit (*Guten Morgen, Guten Tag, Guten Abend*), oder umfangreicher sein. So werden bei einem großen Verkaufsportale dem Kunden für jedes Produkt, dass er anklickt, etliche Informationen angezeigt, ausgelöst durch besagtes Anklicken. Diese Informationen sind in einer riesigen Datenbank (einem digitalen Karteikasten mit ebensolchen Karteikarten) gespeichert, aus der sie bei jedem Aufruf herausgeholt und angezeigt werden. Diese Leistung (Herausholen und Anzeigen) erbringen dynamische Webseiten mit Hilfe von **Skriptsprachen** ([PHP](#), [JavaScript](#), [Python](#)).

Eine von diesen Sprachen lernen Sie im Unterricht kennen, zugleich mit den **Grundlagen des Programmierens**.

3. Ein **Modellrechner** ist ein Programm, das die Arbeitsweise eines Rechners veranschaulicht: Es macht den 'Motor' des Rechners, den Prozessor, sichtbar und verständlich, ähnlich wie ein Modell anderer technischer Geräte (Automotor, Elektronenmikroskop, Wasserkraftwerk) diese durchschaubar macht.

Informatik

Algorithmen I und Sequentielle Datenstrukturen (E2)

1. [Algorithmen](#) sind Handlungsanweisungen zur schrittweisen Lösung eines Problems. Im Alltag (ver)stecken sie (sich) in Koch- oder Backrezepten, Bedienungsanleitungen für technische Geräte (Fotokamera, Mikrowelle), Anleitungen für Aufbau oder Inbetriebnahme von Geräten (Ikea-Möbel, WLAN-Router). Alle diese Beispiele haben eine gemeinsame Form: *Das, was man zur Erreichung eines bestimmten Zwecks tun soll, wird in einer geordneten Abfolge von Schritten möglichst genau beschrieben* (1. Tue dies. 2. Tue das. 3. Tue jenes).

Hier lernen Sie, wie man Algorithmen (Handlungsanweisungen zur Lösung eines Problems) entwirft und diese anschließend in eine Programmiersprache ([Python](#)) übersetzt, damit die eigentliche Problemlösung von der Maschine erbracht werden kann.

2. **Sequenzielle Datenstrukturen.** Für die oben erwähnten 'Problemlösungen' (Kochen, Backen, Gerät benutzen) braucht man Zutaten. Bei den Algorithmen für Computer heißen die Zutaten *Datentypen* (z.B. Zahlen, Buchstaben) und *Datenstrukturen* (z.B. Listen, Bäume).

Sequenzielle Datenstrukturen sind solche, die Elemente wie Perlen an einer Kette aufreihen, z.B. Wörter oder Texte: Wörter verketteten Buchstaben, Texte verketteten Wörter und Satzzeichen. Das Paradebeispiel für sequenzielle Datenstrukturen sind **Listen**.

Typische Probleme für die Arbeit mit solchen Datenstrukturen sind [Suchen](#) (z.B. ein Wort in einem Text, eine Telefonnummer in einem Adressbuch) und [Sortieren](#) (z.B. der Einträge eines Adressbuchs).

Qualifikationsphase

Der Informatikunterricht in der Q-Phase setzt die Kenntnisse aus E2 voraus.

Was macht man im Informatikunterricht in der Q-Phase?

A. Kurze Antwort

Informatiker

- analysieren Probleme,
- konstruieren und formalisieren Lösungsverfahren (Algorithmen), mit denen man diese Probleme lösen kann,
- konstruieren ein Modell eines Ausschnitts der Wirklichkeit und
- übersetzen schließlich Problemlösung oder Modell in eine Programmiersprache

Informatik

Die Semesterthemen für Q1 bis Q4 lauten:

1. Algorithmen II und [Objektorientierte Programmierung](#) (mit Python)
2. [Datenbanken](#) (mit SQL und Python)
3. Eines der Themen aus der folgenden Liste:
 - Kryptologie (Verschlüsseln und Codebrechen)
 - Höhere Datenstrukturen (Bäume, Graphen)
 - Theoretische Informatik
 - Neuronale Netze
 - Nebenläufige Prozesse
4. Vertiefung eines Themas aus der vorstehenden Liste oder ein nichtbehandeltes Thema daraus

B. Lange Antwort

Das Wort [Informatik](#) ist eine Zusammensetzung aus *Information* und *Automatik*, man könnte für den Unterricht auch sagen: *Information* und *Mathematik*. Sie sollten das Fach Informatik nur dann wählen, wenn Sie relativ gute Kenntnisse und Fertigkeiten in Mathematik mitbringen.

In beiden Fachgebieten, Mathematik und Informatik, geht es darum **Probleme zu lösen**, oder einen Sachverhalt zu **analysieren** und daraus ein *mathematisches oder informatisches Modell* zu **konstruieren**.

- *Mathematik*: Wie löst man eine Gleichung? Wie ein Gleichungssystem? Wie formuliert man einen Sachverhalt als Gleichung? (die berüchtigten Textaufgaben)
- *Informatik*: Wie sortiert man eine riesige Datenmenge? Wie sucht man etwas – ein Wort in einem Text, eine Datei auf der Festplatte? Wie steuert man etwas? – den Verkehrsfluss in einer Stadt, oder eine Raumfähre?

(Mit Suchen, Sortieren und Steuern verbringen die meisten Computer dieser Welt einen Großteil ihrer Zeit.)

Es gibt allerdings einen wesentlichen Unterschied: In Mathematik löst ein Mensch die Probleme selbst, indem er ein Lösungsverfahren oder eine Formel entwickelt und anwendet. In Informatik überträgt ein Mensch die Problemlösung einer Maschine, dem Computer. Dazu muß er

1. Das Problem verstehen (z.B. Was ist ein Sudoku? Wie löst man es?)
2. Das Problem selbst lösen können (ein Sudoku, besser noch: jedes)
3. Das Lösungsverfahren der Maschine 'verklickern'.

Zu diesem Zweck überträgt er das von Menschen gefundenen Lösungsverfahren – [Algorithmus](#) genannt – in eine Form, die die Maschine zur automatischen (d.h. selbständigen) Problemlösung instand setzt. Die Frage oben sollte also nicht lauten *Wie sucht man etwas?*, sondern vielmehr: *Wie sucht Maschine mit Hilfe von Mensch etwas?* oder: *Wie befähigt der Mensch die Maschine zum Suchen?*

Informatik

Noch ein Beispiel zur Verdeutlichung des Unterschieds: In einer Tabellenkalkulation kann man eine Liste von Einträgen sortieren lassen. *Im Informatikunterricht lernen Sie, wie man – Sie selbst – die Maschine dazu bringt, dass sie es von alleine (d.h. automatisch) tut.* Dazu lernen Sie eine Reihe **maschineller Sortierverfahren** kennen, hier eine Handvoll: *Bubblesort, Selectionsort, Insertionsort, Mergesort, Quicksort.* Sie sehen: Auch Englisch ist für Informatik wichtig, die jeweils aktuelle Python-Dokumentation etwa ist auf Englisch abgefasst, auf Deutsch gibt es nur eine ältere Version.

Aber Informatiker lösen nicht nur Such- oder Sortier-Probleme, sondern auch **Steuer-Probleme** von technischen Prozessen (Verkehrsfluß, Chemieanlage) oder Verwaltungsvorgängen (Geschäftsprozesse einer Bank, Versicherung oder Bibliothek).

Informatiker lösen solche Probleme durch **Objektorientierte Programmierung** oder **Datenbanken**: Das bedeutet:

- sie **analysieren** dazu einen Ausschnitt der Wirklichkeit (z.B. eine Bücherei): Aus welchen Dingen oder Objekten besteht dieser Ausschnitt? (Benutzer, Medien) Wie spielen diese Objekte miteinander zusammen, welche Beziehungen bestehen zwischen ihnen? (Benutzer leihen Bücher aus, müssen eventuell gemahnt werden, etc.). Objekte und Beziehungen stellt man in einem **Objekt-Beziehungs-Diagramm** dar, im Informatik-Jargon ER-Diagramm genannt.
- sie *konstruieren* ein **informatisches Modell**
- sie gießen dieses Modell in eine andere Form – in eine **Programmiersprache**. *Implementieren* nennt man dieses Umgießen oder Übersetzen in eine Programmiersprache.

Diese drei Schritte nennt man kurz **OOA, OOM** und **OOP**, oft zusammengefasst durch den letzten Ausdruck (OOP). Die Abkürzungen bedeuten:

- OOA = Objektorientierte Analyse
- OOM = Objektorientierte Modellierung
- OOP = Objektorientierte Programmierung

Lust auf Informatik bekommen?

Wolfgang Büchel