

2. Algorithmen und Methoden

- Vertiefende Diskussion über Algorithmen
 - Definition
 - Eigenschaften
 - Klassifizierung
- Algorithmen und Methoden in Java
 - Variablen
 - Einfache Anweisungen
- Kontrollstrukturen in Java
 - Anweisungsblock (Sequenz)
 - Selektion
 - Wiederholung
 - Rekursion



Al-Chwarizmi,
der Namensgeber
für *Algorithmus*

Motivation

- Algorithmen sorgen dafür, dass
 - Aufgaben effizienter erledigt werden können,
 - mathematische Probleme einfacher gelöst werden können,
 - Amazon Ihnen passende Produktvorschläge macht,
 - Sie den Weg zum Hörsaalgebäude finden.

Motivation

- Algorithmen lösen wichtige Probleme.
 - Sortieren
 - Eines der bekanntesten Probleme der Informatik.
 - Computer beschäftigen sich sehr oft mit dem Sortieren von Daten.
 - Berechnung des größten gemeinsamen Teilers
 - Euklid hat bereits dazu vor 2300 Jahren einen Algorithmus entwickelt.
 - Ranking von Web-Seiten
 - Algorithmus von Brin & Page wird in Suchmaschinen verwendet.
 - Verschlüsselung von Nachrichten
 - Verwendung von Algorithmen bereits im antiken Rom durch Caesar



Algorithmen im Alltag

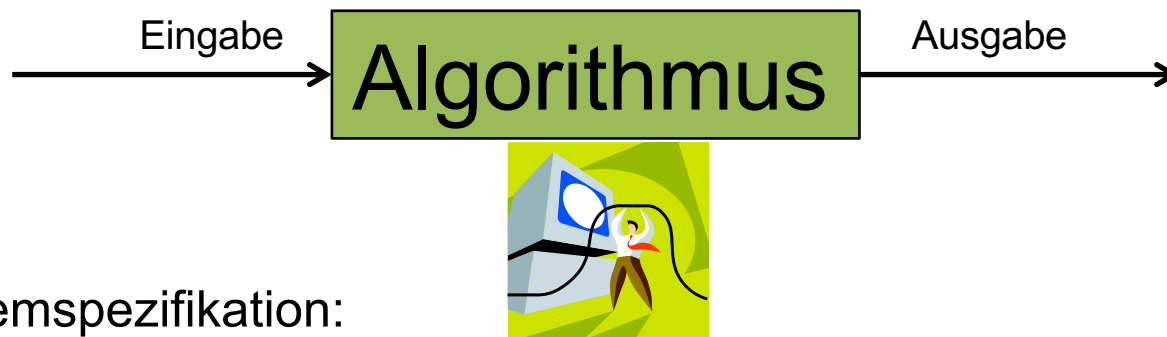
- Das Möbelhaus Ikea bietet Online viele [Montageanleitungen](#) an.
- Jede Gebrauchsanleitung enthält Algorithmen.
 - Wie kann ich mit meiner Kaffeemaschine Kaffee kochen?
 - Wie kann ich die Kaffeemaschine entkalken?
- Wie kann ich einen Router konfigurieren?

Algorithmen und Programme

- Algorithmen können in verschiedener Art und Weise beschrieben werden.
 - So kann z. B. ein Algorithmus mit Hilfe von einer Programmiersprache formuliert werden.
 - Ein **Algorithmus wird als Methode** implementiert, die **Teil eines Programms** ist. Das Programm kann dann auf einem Rechner ausgeführt werden.
 - Viele Algorithmen werden jedoch informell beschrieben, um dann vom Menschen ausgeführt zu werden.
 - Beispiele
 - Algorithmus zum [Aufbauen von Hemnes mit 8 Schubladen](#).
 - Algorithmus zum Entkalken der Kaffeemaschine
 - Algorithmus zum Kochen einer Tasse Kaffee

2.1 Algorithmen und ihre Spezifikationen

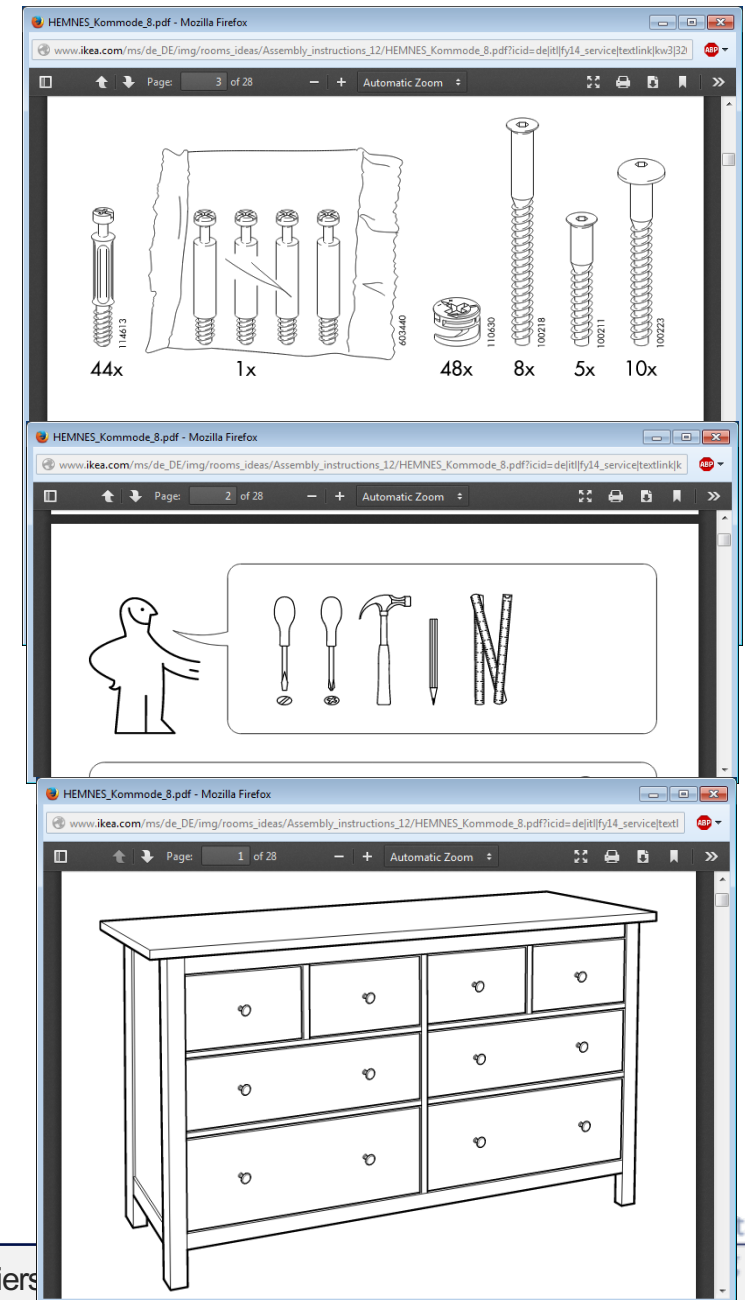
- Algorithmen sind **allgemeine Lösungsverfahren** für eine Problemklasse
 - Der Algorithmus bekommt Daten als Eingabe und liefert ein Ergebnis als Ausgabe.



- Problemspezifikation:
ist eine **vollständige, detaillierte und unzweideutige Problembeschreibung**.
 - vollständig:**
Angabe aller Rahmenbedingungen (**Eingabe**)
 - detailliert:**
Voraussetzungen über Hilfsmittel und Grundoperationen (**Prozessor**)
 - unzweideutig:**
klare Beschreibung, was der Algorithmus tun soll (**Ausgabe**).

Beispiel (Hemnes von Ikea)

- **vollständig:**
 - Es wird genau beschrieben, welche Teile in der Kiste sein sollen.
 - Alle Schrauben sind genau beschrieben.
- **detailliert:**
 - Benötigte Werkzeuge werden gezeigt.
- **unzweideutig:**
 - Eine Zeichnung zeigt das Endprodukt.



Formale Beschreibungsverfahren zur Spezifikationen von Problemen

- **Informelle** in Umgangssprache formulierte **Spezifikationen** genügen i.a. nicht den obigen Kriterien.



- **Spezifikationen** können in der Informatik ganz **formal** in speziellen, auf der Logik basierenden Spezifikationssprachen ausgedrückt werden.
 - Beispiele für formale **Spezifikationssprachen**: Z Notation, VDM

Beispiel: Suche nach einer Telefonnummer

- Informelle Beschreibung: Suche zu einem beliebigen Namen die zugehörige Telefonnummer.

Offene Fragen:

- Vollständigkeit:
 - Wie setzt sich ein Name zusammen?
 - Buchstaben des deutschen Alphabets
 - Sind auch Ziffern und Sonderzeichen erlaubt ? Beispiel „1&1“
 - Besteht ein Name aus Vor- und Nachname?
- Detailliertheit:
 - Welche Hilfsmittel stehen zur Verfügung?
 - Internet
 - Telefonbuch
 - CD-ROM
 - Was bieten die einzelnen Hilfsmittel an Funktionalität?
- Unzweideutigkeit:
 - Wie soll die Ausgabe meiner Suche aussehen?
 - Was passiert, wenn mehrere Telefonnummern gefunden werden?

Beispiel: ggT (größter gemeinsamer Teiler)

- Informelle Problembeschreibung: Für beliebige Zahlen M und N berechne den größten gemeinsamen Teiler

Offene Fragen:

- Vollständigkeit:
 - Welche Zahlen sind zugelassen?
 - Ganze Zahlen, rationale Zahlen, oder sonstige?
 - Dürfen M und N auch 0 sein?
- Detailliertheit:
 - Welche Operationen sind auf den ganzen Zahlen erlaubt?
 - $+$, $-$ oder auch $/$ (ganzzahlige Division), $\%$ (Divisionsrest)?
- Unzweideutigkeit:
 - Was bedeutet „ggT“ überhaupt?
 - Welche Zahlen sind als Ergebnis zugelassen?

Vorbedingung - Nachbedingung

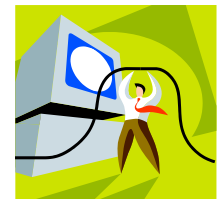
- Probleme werden formal über ein Paar $\{P\} \{Q\}$ spezifiziert.
 - P (**Vorbedingung**)
 - Alle relevanten Eigenschaften, die anfangs vor der Ausführung des Algorithmus gelten.
 - Q (**Nachbedingung**)
 - Alle relevanten Eigenschaften, die am Ende des Algorithmus erfüllt sind.
- P und Q können als Boolesche Funktionen aufgefasst werden, die nur zwei Werte („wahr“ und „falsch“) als Resultat zulassen.

Beispiel (ggt)

- Vorbedingung P
 - M und N sind ganze Zahlen mit $M > 0$ und $N > 0$.
- Nachbedingung Q
 - Das Ergebnis z aus den ganzen Zahlen mit $z > 0$ erfüllt folgende Bedingungen:
 - z ist Teiler von M
 - z ist Teiler von N
 - für jede ganze Zahl y, die N und M teilt, gilt $y \leq z$

Algorithmus – wichtige Begriffe

- Definition
 - Ein **Algorithmus** ist eine präzise und endliche **Beschreibung** eines allgemeinen Verfahrens zur **schrittweisen Lösung** eines **Problems**. Der Algorithmus besteht aus einzelnen **Schritten** und **Vorschriften**, welche die **Ausführung der Schritte** kontrollieren.
 - Jeder Schritt muss
 - **klar** und **eindeutig** beschrieben sein und
 - mit **endlichem Aufwand** in **endlicher Zeit** ausführbar sein.
- Ein Algorithmus wird durch einen **Prozessor** ausgeführt.
- Ein (sequentieller) **Prozess** bezeichnet die sequentielle Folge von Ausführungsschritten, die durch einen Prozessor bei der Verarbeitung eines Algorithmus erzeugt wird.




Gegenbeispiele: Was ist kein Algorithmus!

- Beschreibung durch **Analogien**
Das Zerteilen von Fliesen läuft im Prinzip so wie das Zerteilen eines Ziegels.
- Beschreibung durch **Beispiele**
Berechne die ersten 10 Zahlen der Folge 2, 3, 5, 7, 11,
 - Was ist die nächste Zahl der Folge?
- Verwendung **nicht-ausführbarer und nicht-Grundoperationen**
 - Laufe die nächsten 100 Meter in 5 Sekunden.
 - Zur besseren Anbindung des Campus an die Stadt, soll ab dem WS 2018/19 ein Helikopter verwendet werden.
- **Verletzung der Allgemeinheit** (Problem: Addieren zweier Zahlen)
 - 2 und 3 ergibt zusammen 5
 - Ein **Algorithmus** lässt sich **stets auf eine Klasse von Problemstellungen** anwenden (z.B. die Addition zweier Zahlen).
 - Jeder Algorithmus besitzt deshalb **Eingabeparameter**, die beim Start eines Prozesses **mit konkreten Werten zu belegen** sind.

Korrektheit von Algorithmen

- Zwingend notwendig ist die **Korrektheit**, d.h. der Algorithmus muss tatsächlich die **Spezifikation $\{P\}\{Q\}$ erfüllen**:
 - Für **jeden möglichen Prozess** eines Algorithmus muss folgender Sachverhalt gelten:
 - Falls beim Start des Prozesses die **Funktion P** wahr liefert,
 - muss auch am Ende des Prozesses die **Funktion Q** wahr liefern
 - Erfüllt ein Algorithmus A die Spezifikation $\{P\}\{Q\}$, so schreiben wir auch
 $\{P\} A \{Q\}$
 - Die **Vorbedingung P** ist abhängig von den **Eingabe** und die **Nachbedingung Q** ist abhängig von der **Ausgabe** des Algorithmus A.



„Jetzt hat man jedoch beim Nachfolgemodell Ariane 5 nicht beachtet, dass die Ariane 5 eine schnellere Rakete ist. In dem Moment sind die Daten, die übermittelt worden sind, höhere Zahlen gewesen. Und diese Zahlen haben in den Bereich, den der Computer auffassen konnte, nicht mehr reingepasst.“

Nicht-funktionale Eigenschaften von Algorithmen

- Algorithmen sollten einfach zu **verstehen** sein.
 - Erläuterung des Prinzips durch ein Beispiel
 - **Zusätzliche Beschreibung in Form von *Dokumentation***
- **Laufzeit-** und **Platzbedarf** sollten möglichst niedrig sein.
 - Effizienzanalyse von Algorithmen
 - Experimentelle Validierung der Kosten

Terminierung von Algorithmen

Definition (**Terminierung**)

- Ein **Prozess** terminiert, wenn er aus einer **endlichen Anzahl von Ausführungsschritten** besteht.
- Ein **Algorithmus** terminiert, wenn **jeder seiner möglichen Prozesse terminiert**.
- Bemerkungen
 - Oft ist „Algorithmus“ in der Literatur so definiert, dass die **Terminierung** bereits **inhärent gefordert** wird (z.B. Suche nach Webseiten).
 - nicht-terminierende Algorithmen sind aber **auch von Interesse**
 - Ampelsteuerung
 - Verarbeitung von Daten eines Sensors oder Tickers:
Kontinuierliche Berechnung des durchschnittlichen Börsenwertes einer Aktie innerhalb der letzten sieben Tage.

Deterministische Algorithmen

- **Determinismus** bezieht sich auf den Prozess
 - Ein Algorithmus ist **deterministisch**, wenn es zu jeder möglichen Eingabe genau einen Prozess gibt.
 - Kann bei nichtdeterministischen Algorithmen den zu einer Eingabe gehörenden Prozessen eine Wahrscheinlichkeit zugeordnet werden, so spricht man auch von **stochastischen Algorithmen**.
- **Beispiele** für nichtdeterministische Algorithmen
 - Spielzüge bei einem Schachspiel
 - Verkehrsregelung an einer Kreuzung

Determiniertheit von Algorithmen

- **Determiniertheit** bezieht sich auf das **Ergebnis** eines Algorithmus
 - Ein Algorithmus ist **determiniert**, wenn die **Eingabe** das **Resultat** des Algorithmus **eindeutig bestimmt**.
- Beispiele für determinierte Algorithmen
 - Aufbauanleitung bei Ikea
 - Größter gemeinsamer Teiler von zwei Zahlen
- Beispiele für nichtdeterminierte Algorithmen
 - Spielzüge bei Spielen mit Würfeln
 - Approximative Suchverfahren für Probleme, zu denen es keinen schnellen Algorithmus gibt.

Sequentielle und parallele Algorithmen

- Diese Begriffe **beziehen sich auf die Prozesse**.
 - Ein Prozess heißt **parallel**, wenn **Einzelschritte gleichzeitig** ausgeführt werden.
 - Ein Algorithmus heißt **parallel**, wenn er **parallele Prozesse zulässt**.
Andernfalls heißt er **sequentiell**.
- Beispiele:
 - Beim **Hausbau** kann eine Wand aus Ziegeln gemauert werden und gleichzeitig können die Ziegel zugeschnitten werden.
 - Bei der **Berechnung von $a^2 + b^2$** kann die Quadratberechnung der Zahlen a und b parallel zueinander ablaufen.
 - **Suchen in Telefonbüchern** kann parallel erfolgen (wenn es mehrere Telefonbücher gibt, ansonsten ist Parallelität nicht sehr effektiv)

Zusammenfassung

- Formale Spezifikation von Problemen
 - Vorbedingung und Nachbedingung
- Algorithmus
- Korrektheit von Algorithmen
- Nicht-funktionale Eigenschaften
 - Laufzeit und Speicherverbrauch eines Algorithmus
- Arten von Algorithmen
 - Deterministisch
 - Determiniert
 - Sequentielle und parallele Algorithmen