



BLAT 2020 | BarCamp

Computational Thinking –
ein Konzept zum fächerübergreifenden Aufbau
von Problemlösestrategie

Auszug der Vortragsfolien

Raphael Fehrmann | WWU Münster

Was Europa über Algorithmen weiß und denkt

- „Die Menschen in Europa wissen wenig über Algorithmen.“
- **48 Prozent der europäischen Bevölkerung wissen nicht, was ein Algorithmus ist.**
- Auch ist weniger als der Hälften der europäischen Bevölkerung bekannt, dass Algorithmen bereits in vielen Lebensbereichen eingesetzt werden. Besonders niedrig sind dabei die Werte für Anwendungsfelder, wo die Entscheidungen von Algorithmen potenziell folgenreich für die soziale Teilhabe sind, etwa bei der Kreditvergabe, der Bewerberauswahl und der medizinischen Diagnostik. [...]
- So assoziieren Europäerinnen und Europäer mit Algorithmen sowohl positive Aspekte wie Effizienz und Zeitersparnis als auch Negatives wie Angst oder das Risiko der Manipulation. Insgesamt **überwiegt die optimistische Grundhaltung.“**

Definition: Algorithmus

Unter einem **Algorithmus** versteht man

- ein **Verfahren**, welches **eindeutig, endlich beschreibbar** und **mechanisch** ist und
- der **Lösung eines vordefinierten Problems** – bzw. der Lösung mehrerer ähnlicher Probleme, die derselben Problemklasse zugehören – dient. Hierfür umfasst der Algorithmus eine Reihe **präziser Anweisungen**.
- Zu jedem **Zeitpunkt des Verfahrens** muss der **Folgeschritt eindeutig** durch den vorangegangenen Schritt festgelegt.
- Nach der Eingabe der jeweiligen Daten und der Ausführung des Algorithmus **bricht das Verfahren nach endlich vielen Schritten** ab, es liefert das gesuchte Ergebnis, die Lösung des Problems.

Beispiele aus der digitalen Anwendung:

- Partnersuche
- Google-Suche
- Werbeeinblendung
- Navigation / Wegeberechnung
(Dijkstra-Algorithmus, kürzester Weg)
- Bewerberauswahl
- Ampelschaltung
- ...

Gibt es evtl. Vorgehensweisen im Alltag, die Ähnlichkeiten zu algorithmischen Vorgehensweisen aufweisen, sich aber bzgl. der Eigenschaften ggf. unterscheiden?

Aktivitäten wie **Kochen, Backen, Zähneputzen oder der Weg zur Arbeit/Schule** weisen ebenfalls klar strukturierte Handlungsvorschriften auf.

Diese müssen und werden allerdings oftmals nicht ganz so genau den genannten Eigenschaften von Algorithmen entsprechen, sondern können an einigen Stellen auch abweichen und funktionieren trotzdem noch.



Anwendungsbeispiel:
<https://idea-instructions.com/quick-sort/>

- basierend auf dem informatischen Quick Sort
- ungeordnete Liste von Türmen (Zahlen), Problem: aufsteigend sortieren
- Algorithmisches Vorgehen:
 - Start: Würfeln zur Auswahl eines Startturms – hierdurch Einteilung der Elemente in zwei Teilmengen
 - Linie auf der Höhe des Turms zeichnen
 - Steine, die höher als Linie sind: kennzeichnen mit „nach rechts verschieben“
 - Steine, die niedriger als Linie sind: kennzeichnen mit „nach links verschieben“
 - Sortiere
 - Schleife: wiederhole und durchlaufe Schema für beide Teilmengen so lange, bis richtig sortiert – für höhere Türme und für niedrigere Türme
 - danach: Zusammenführung der Ergebnisse

Das Dagstuhl-Dreieck



2019: Erweiterung des
Dagstuhl-Dreiecks auf ein
multidimensionales Modell
mit mehreren Ebenen
(„Frankfurter Dreieck“),
zum Weiterlesen: Weich
2019

Grafik: R. Fehrman

In Anlehnung an: Döbeli Honegger 2017a, GI 2016, 2019

computational thinking – informatisches, problemorientiertes Denken (J. Wing)

- computational thinking als eine „Reihe von Gedankenprozessen, die an der Formulierung und Lösung von Problemen beteiligt sind“ (Bollin 2016, S. 28)
- mit dem Ziel, diese Reihe / diesen Algorithmus so darzustellen, dass ein Computer diese/n ausführen könnte (ebd.)

Annahme:

- Informatische Prinzipien finden sich in allen Lebensbereichen.
- Verwendung der informatischen Komponente nicht im traditionellen Sinne (Codierung von Informationen in Zeichenfolgen),
- sondern als **Entwickeln einer Problemlösekompetenz durch algorithmisch-schematisches Handeln** (Probleme identifizieren bzw. genau definieren, Strategien entwickeln, Lösungen dokumentieren und kommunizieren, Kreativität fördern etc.)

vgl. Wing 2006
vgl. Bollin 2016

Digitale Souveränität erfordert computational thinking

computational thinking – Definition

Ausprägung der analytischen Fähigkeiten zur Dekomposition und Abstraktion, welche durch die kleinschrittige Zerlegung und Analyse von bspw. informatischen Problemstellungen zur Lösung dieser erfahren werden

vgl. Wing 2006, S. 33

- fokussiert die Ausprägung der analytischen Fähigkeiten zur Dekomposition und Abstraktion,
- Ausbildung durch die kleinschrittige Zerlegung und Analyse von bspw. informatischen Problemstellungen zur Lösung

Digitale Souveränität erfordert computational thinking

computational thinking betont

- **Konzepte der Problemlösung** (Logik / Analysieren, Abstraktion / Auswahl des Relevanten, Dekomposition / Zergliedern, Algorithmisierung / Nachvollziehen von Abläufen)
- und **Handlungsweisen** (kreatives Gestalten, Debuggen / Fehleranalyse und -behebung, Durchhaltevermögen, Kollaboration),

die fächerübergreifend das lebensweltliche Handeln eines Jeden prägen

vgl. Wing 2006

vgl. Böllin 2016

vgl. Baumann 2016

Algorithmisches Denken findet in zahlreichen Handlungsfeldern

(auch ohne den Einsatz von Informations- und Kommunikationstechnik) Anwendung!

vgl. Romeike 2017

vgl. Futschek 2016

Dreischritt des computational thinkings:

- Formulierung des Problems (Abstraktion)
- Formulierung der Lösungsschritte (Automatisierung)
- Ausführung und Auswertung der Lösungsschritte (Analyse)

Beispiel: Spiel „Wer bin ich?“

vgl. Wing 2006

vgl. Böllin 2016

vgl. Baumann 2016

Digitale Souveränität erfordert computational thinking

- **Resultierender schulischer Bildungsauftrag:**

- **fächerübergreifender Aufbau** eines „Verständnis[es] für die Nutzung und Bearbeitung von Daten und das **Lösen von Problemen** [u.a.] mithilfe computergestützter Methoden“
(Hartmann et al. 2015, S. 75 f.)
- Förderung des **problemlöseorientierten Denkens** in „abstrakten Modellen und in vernetzten Systemen“ (ebd.)
- exemplarisches **Erstellen von Modellen und Nutzung von Simulationen** zur Planung und Entscheidungsfindung inkl. der Ableitung von Grenzen dieser Modelle
- didaktisch **reduzierter Aufbau von Problemlösekopetenz: Coding von Lernrobotern**

vgl. Hartmann et al. 2015, S. 75 f.

Handlungen beim Aufstellen von Algorithmen

- Definieren der Ausgangslage und dem Ziel
- Entwickeln von möglichen Teilschritten, Abbildung dieser
- Formalisieren und Festhalten
- Bewerten der bisherigen Erkenntnisse, Überprüfung auf Korrektheit
- Testen anhand von Beispieldaten / Verifizieren bzw. das Begründen der korrekten Ausführung an bekannten Bedingungen
- Anwenden, Bewerten, Korrigieren, Verbessern, Weiterentwickeln inkl. der Überprüfung auf Effektivität

Programmieren bzw. algorithmisches Denken als Problemlösen

Der „computational thinker“



Ist mein Lösungsweg der effizienteste Weg?
Habe ich die schnellste Lösung gefunden?
Braucht mein Lösungsweg die wenigsten Ressourcen?
Führt mein Lösungsweg zur richtigen Antwort?
Ist mein Lösungsweg anwendbar auf weitere Probleme?

Programmieren bzw. algorithmisches Denken als Problemlösen

Programmieren, Problemlösen, Modellieren – comp. thinking?!

- **Computational Thinking / algorithmisches Denken als Problemlösen**
- **Ziel: Verständnis entwickeln, ohne Programmieren zu können**
(vgl. auch Studienergebnisse aus ICILS 2018 / Eickelmann et al., 2019)

Das Konzept des **Computational Thinkings** lässt sich laut der Strategie der Kultusministerkonferenz 2016 „**Bildung in der digitalen Welt**“ in folgenden **Kompetenzbereichen** verorten:

- **5. Problemlösen und Handeln,**
insb. 5.1. Technische Probleme lösen, 5.5. Algorithmen erkennen und formulieren
sowie auf Basis des Bereichs 5.:

- **1. Suchen, Verarbeiten und Aufbewahren,** insb. 1.2. Auswerten und Bewerten
- **2. Kommunizieren und Kooperieren,** insb. 2.5. An der Gesellschaft aktiv teilhaben
- **4. Schützen und sicher Agieren,** insb. 4.1. Sicher in digitalen Umgebungen agieren
- **6. Analysieren und Reflektieren,** insb. 6.2. Medien in der digitalen Welt verstehen u. refl.

Computational Thinking

Umsetzungsideen

... plugged & unplugged

- „mit Spielen, Zaubertricks und Rätseln“ (s. Literaturempfehlung)
- durch Analyse von alltäglichen algorithmenähnlichen Handlungen (Zähneputzen, Kochanleitungen, Einpacken eines Geschenks) nach dem Prinzip des CTs
- durch das Hinterfragen digitaler Phänomene (personalisierte Werbeeinblendungen, RFID-Warensicherung, QR-Codes, Barcodes,...)
- durch die Modellierung einfacher digitaler Systeme (Funktionsweise einer Straßenampel in reduzierter Form)

...mithilfe von Lernrobotern

- Lernroboter im Unterricht

www.wwu.de/Lernroboter

(mit Unterrichtsmaterialien ab ca. 11/2020)



Computational Thinking

Umsetzungsideen

Raum für Diskussionen

Wie Algorithmen wirken...

This person doesn't exist – ein Grafikgenerator entwirft menschliche Fotos

<https://www.thispersondoesnotexist.com/>



P. Curzon, P. W. McOwan (2018): **Computational Thinking: Die Welt des algorithmischen Denkens – in Spielen, Zaubertricks und Rätseln.** Berlin: Springer.

Hartmann, Werner & Hundertpfund, Alois (2015): **Digitale Kompetenz – Was die Schule dazu beitragen kann.** Bern: hep Verlag AG.

Döbeli Honegger, Beat (2017): **Mehr als 0 und 1 – Schule in einer digitalisierten Welt.** 2. Auflage. Bern: hep Verlag AG.

Literaturverzeichnis



- Baumann, Wilfried (2016): Pladoyer für Computational Thinking. In: OCG Journal (02), S. 13. Online verfügbar unter <https://www.ocg.at/sites/ocg.at/files/medien/pdfs/OCG-Journal1602.pdf>, Tag des letzten Zugriffs: 14.01.2020.
- Bollin, Andreas (2016): COOLe Informatik. In: OCG Journal (02), S. 28. Online verfügbar unter <https://www.ocg.at/sites/ocg.at/files/medien/pdfs/OCG-Journal1602.pdf>, Tag des letzten Zugriffs: 14.01.2020.
- Döbeli Honegger, Beat (2017a): Mehr als 0 und 1 – Schule in einer digitalisierten Welt. 2. Auflage. Bern: hep Verlag AG.
- Eickelmann, Birgit; Bos, Wilfried; Gerick, Julia; Goldhammer, Frank; Schaumburg, Heike; Schwippert, Knut; Senkbeil, Martin & Vahrenhold, Jan (2019): ICILS 2018 Deutschland - Computer- und informationsbezogene Kompetenzen von Schülerinnen und Schülern im zweiten internationalen Vergleich und Kompetenzen im Bereich Computational Thinking. Münster: Waxmann. Zugriff über URL: https://kw.unipaderborn.de/fileadmin/fakultaet/institute/erziehungswissenschaft/Schulpaedagogik/ICILS_2018__Deutschland_Berichtsbund.pdf, Tag des letzten Zugriffs: 12.11.2019.
- Futschek, Gerald (2016): Bildung 4.0 : Informatisches Denken ist Schlüsselkompetenz. In: OCG Journal (02), S. 20. Online verfügbar unter <https://www.ocg.at/sites/ocg.at/files/medien/pdfs/OCG-Journal1602.pdf>, Tag des letzten Zugriffs: 14.01.2020.
- Gesellschaft für Informatik e. V. (GI, 2016): Dagstuhl-Erklärung – Bildung in der digitalen vernetzten Welt. Eine gemeinsame Erklärung der Teilnehmerinnen und Teilnehmer des Seminars auf Schloss Dagstuhl. Online-Bezug über URL: https://gi.de/fileadmin/GI/Hauptseite/Themen/Dagstuhl-Erkla_rung_2016-03-23.pdf, Tag des letzten Zugriffs: 27.07.2018.
- Gesellschaft für Informatik e. V. (GI, 2019): Frankfurt-Dreieck zur Bildung in der digital vernetzten Welt – Ein interdisziplinäres Modell. Online-Bezug über URL: <https://dagstuhl.gi.de/fileadmin/GI/Allgemein/PDF/Frankfurt-Dreieck-zur-Bildung-in-der-digitalen-Welt.pdf>, Tag des letzten Zugriffs: 16.11.2019.
- Grzymek, Viktoria & Puntschuh, Michael (2019): Was Europa über Algorithmen weiß und denkt. Ergebnisse einer repräsentativen Bevölkerungsumfrage. Hg. v. Bertelsmann Stiftung. Gütersloh. Online-Bezug über URL: <https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/WasEuropaUEberAlgorithmenWeissUndDenkt.pdf>, Tag des letzten Zugriffs: 15.11.2019.
- Hartmann, Werner & Hundertpfund, Alois (2015): Digitale Kompetenz – Was die Schule dazu beitragen kann. Bern: hep Verlag AG.
- Köpp, Sabine; Kuhlen, Britta & Voll, Sabine (2018): Mein Medienheft 3 / 4 – Digitale Medien. Stuttgart, Leipzig: Ernst Klett.
- Kreis Soest, Medienzentrum (2018): Methodenkatalog zum Medienkompetenzrahmen NRW. Soest: Kreis Soest. Online-Bezug über URL: https://www.kreis-soest.de/bildung_integration/bildung/medienzentrum/materialien/materialien.php.media/386821/Methodenkatalog_web.pdf, Tag des letzten Zugriffs: 31.10.2019.
- Meyer, Manfred & Neppert, Burkhard (2012): Java. Algorithmen und Datenstrukturen; mit einer Einführung in die funktionale Programmiersprache Clojure. Herdecke: W3L-Verl. Das verwendete Kapitel 3 kann über den Springer-Verlag als Leseprobe (PDF) bezogen werden – Bezug über URL: https://www.springer-campus-it-onlinestudium.de/w3lmedia/W3L/Medium224171/9783937137179_Leseprobe.pdf, Tag des letzten Zugriffs: 20.11.2019.
- Romeike, Ralf (2017): Wie informative Bildung hilft, die digitale Gesellschaft zu verstehen und mitzugestalten. In: Eder, Sabine; Mikat, Claudia; Tillmann, Angela (Hrsg.): Software takes command – Herausforderungen der „Datafizierung“ für die Medienpädagogik, in: Theorie und Praxis, S. 105-118. München: kopaed. Bezug über URL: https://computingeducation.de/pub/2017_Romeike_GMK2016.pdf, Tag des letzten Zugriffs: 15.11.2019.
- Wing, Jeannette Marie (2006): Computational Thinking - It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. In: Communicatio nof the ACM 49.3, 05/2006, S. 33-35. Bezug über URL: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>, Tag des letzten Zugriffs: 27.07.2018.

Raphael Fehrman

Promovend am Institut für Erziehungswissenschaft der

Westfälischen Wilhelms-Universität Münster

Forschungsprojekt »Lernroboter im Unterricht«

Tel.: 0251 83-32185 | E-Mail: raphael.fehrman@wwu.de

Web: www.wwu.de/Lernroboter/rf