

Coding, Informatisches Denken und problemlösendes Denken im C4

Konzept zur Förderung der Kompetenzen im Bereich des Computational Thinking

Begriffserklärungen

In EU-Ländern gibt es bereits viele Initiativen mit unterschiedlichen Ansätzen zur Einbindung von Computational Thinking in den Lehrplan. Anlässlich des Schulbeginns im September 2019 schrieb Luxemburgs Bildungsminister:

„Le coding et la pensée computationnelle feront partie de l’enseignement de toutes les classes de l’école fondamentale et des classes inférieures de l’enseignement secondaire à partir de la rentrée 2020-2021.

À l’enseignement fondamental, le coding sera intégré en 2020-2021 au cycle 4 dans les cours de mathématiques. En 2021-2022, il sera introduit aux cycles 1 à 3, de manière transversale dans toutes les branches. “

Im Pressebericht werden die beiden Begriffe „Coding“ und „Computational Thinking“ benutzt. Daher vorab Begriffserklärungen:

- *Coding* (Cambridge Dictionary | English, 2019):
“[1] to represent a message in code so that it can only be understood by the person who is meant to receive it”
[2] COMPUTING “specialized: to write computer programs (= instructions)”
- *Computational Thinking* (Wing, 2010, S. 1):
„Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can effectively be carried out by an information-processing agent.”

Es handelt sich nach Wing bei **Computational Thinking** also um

1. einen Gedankenprozess
2. welcher zu einer Lösung eines Problems führen soll
3. die so notiert werden soll, dass ein Dritter (Mensch/Computer/Roboter) diese ausführen kann.

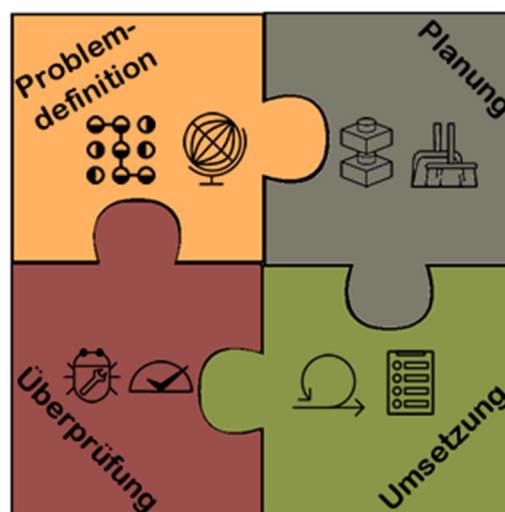
Daraus ergibt sich ein dreistufiger Prozess (vgl. Repenning, 2015) bestehend aus

1. der Problemformulierung, meistens eine präzise Fragestellung, welche auf einer Problemanalyse gründet.
2. einer Repräsentation einer Lösung. Diese kann textbasiert oder grafisch erweitert sein.
3. der Ausführung der Lösungsrepräsentation mit anschließender Evaluation.

Der zweite Begriff zu **Coding** betrifft den Bereich des Computing und zeigt auf, dass Coding in diesem Bereich gleichzusetzen ist mit „Programmieren“, also der Niederschrift einer Handlungsabfolge in einer für die Maschine verständlichen Art und Weise. Computational Thinking oder Informatisches Denken ist also viel breiter gefasst als Coding. Coding konzentriert sich laut Wing auf die Darstellung von Problemen und ihren Lösungen und nach Repenning auf eine geeignete Darstellung der Lösungen, entweder textbasiert oder grafisch aufbereitet.

Etappen und Vorgehen

Computational Thinking oder Informatisches Denken ist eine Voraussetzung, um Coding zu verstehen. Deshalb ist es ideal im Unterricht mit Computational-Thinking-Aktivitäten zu beginnen. Beim Informatischen Denken steht die Aufsetzung einer Problemlösungsstrategie, eines Algorithmus im Vordergrund. Dieser muss so aufbereitet sein, dass ein anderer die Aufgabe ausführen kann. „From doing it to having it done“ bezeichnet Duchâteau (1992) den Prozess. Ziel einer Computational-Thinking-Aktivität ist nicht das Lösen der Aufgabe durch den/die SchülerIn, sondern das Aufsetzen einer Handlungsabfolge, die von einem Menschen oder Computer durchgeführt und zur Lösung gebracht wird. Dieser Prozess besteht aus vier aufeinander folgenden Etappen, welche auf acht Kompetenzen des Computational Thinkings basieren.



1. **Etappe: Problemdefinition**

Das Problem wird untersucht und mit anderen, bereits bearbeiteten, Aufgaben verglichen.

Kernfragen: Gibt es Übereinstimmungen oder gemeinsame Merkmale? Sind gemeinsame Muster mit anderen Aufgaben zu erkennen?

Kompetenzen: Mustererkennung, Verallgemeinerung

2. **Etappe: Planung**

Das Problem wird in mehrere Einzelschritte zerlegt, komplexe Aufgaben werden vereinfacht. Unwichtige Details werden weggelassen. Es gilt den Kern des Problems zu erkennen und diesen kleinschrittig zu definieren.

Kernfragen: Welche unwichtigen Details können weggelassen werden? Was ist der Kern des Problems? Wie kann ich die Aufgabe in Teilschritte zerlegen?

Kompetenzen: Zerlegung, Abstraktion

3. **Etappe: Umsetzung**

Handlungsschritte werden festgelegt. Die nutzende Person überlegt, wie er oder sie die Handlungen definieren kann, damit ein anderer diese ausführen kann. Die Handlungsanweisungen (Algorithmen) müssen präzise und detailliert und in einer bestimmten Reihenfolge vorgegeben sein. Wiederholungen lassen sich in einem Block zusammenfassen, um die Anweisungen übersichtlicher zu gestalten und den Algorithmus zu optimieren.

Kernfragen: Wie können die einzelnen Lösungsschritte notiert werden? Gibt es sich wiederholende Schritte, welche zusammengefasst werden können?

Kompetenzen: Algorithmisches Denken, Wiederholung der Handlungsanweisungen (Iteration)

4. **Etappe: Überprüfung**

Die Handlungsschritte werden umgesetzt: sie können zum vorgegebenen Ziel führen oder auch nicht. Im letzteren Fall muss der Algorithmus überarbeitet, Fehler gefunden und korrigiert werden (Debugging). Die vorher aufgestellten Schritte werden evaluiert. Möglicherweise braucht es einen Neuanfang der ersten Etappe, um das Problem neu anzugehen.

Kernfragen: Wurde die Aufgabe gelöst, kann auch über mögliche Optimierungen nachgedacht werden. Gibt es einfachere, schnellere, elegantere Lösungen?

Kompetenzen: Debugging, Evaluation

Praktische Umsetzung

Das vorliegende Konzept richtet sich primär an LehrerInnen des vierten Zyklus, welche noch gar keine bis wenig Erfahrung mit Computational Thinking und Coding haben.

Zum Schulbeginn 2020 werden folgende Materialien zur Verfügung gestellt:

- Fortbildungen, Hospitationsangebote am IFEN
- Film zur Veranschaulichung von Computational-Thinking-Aktivitäten in der Grundschule
- Webseite mit weiterführenden Erklärungen und Links
- Webseite mit externen Organisationen

Zusätzlich erhalten die Schulen für des Zykel 4 ein Paket mit folgendem Inhalt:

- Ozobots



- Lux-Robo Modikit



- Kniwwelinos



- Begleitheft mit Unterrichtsideen und kurzen Erklärungen

Im Begleitheft werden den Lehrkräften 36 Unterrichtsideen vorgestellt. Diese ermöglichen ihnen, jede Schulwoche eine Aktivität im Bereich des Computational Thinking durchzuführen.

Das Begleitheft sieht folgenden Aufbau vor:

1. **„Unplugged“- Aktivitäten:** Einstieg in das Informatische Denken mit Aktivitäten, die kein digitales Material erfordern. Die SchülerInnen lernen spielerisch Informatisches Denken.
Nähere Informationen unter: <https://csunplugged.org/en/what-is-computer-science/>.
2. **Aktivitäten mit Ozobots:** Diese Roboter ermöglichen den spielerischen Einstieg ins Programmieren mit Hilfe von **Farbcodes**.
Nähere Informationen unter: <https://ozobot-deutschland.de/>.

3. **Blockbasierte Programmiersprache mithilfe von Ozobots:** Die Ozobots werden am Computer oder Tablet mit einer Programmiersprache programmiert.

Nähere Informationen unter: <https://ozoblockly.com/>.

4. **Einsatz von Scratch:** Ist die blockbasierte Sprache bekannt, können kleine Projekte in Scratch leicht umgesetzt werden.

Nähere Informationen unter: <https://scratch.mit.edu/>.

5. **Umgang mit Kniwwelino:** Diese Platine kann mit **Scratch** programmiert und für kreative Aktivitäten genutzt werden.

Nähere Informationen unter: <https://www.kniwwelino.lu/>.

6. **Einsatz von ModiKit:** Diese Hardware ist eine blockbasierte Programmiersprache. Die SchülerInnen lernen eigene Projekte anhand dieser Module aufzubauen.

Nähere Informationen unter: <https://www.kickstarter.com/projects/luxrobo/modi-create-anything-you-want-with-robotics-of-thi?lang=de>.

Das Konzept baut anhand der Unterrichtsideen das Informatische Denken von „Unplugged“-Aktivitäten über Programmieren mit Farbcodes bis hin zum Coden von Platinen und Modulen auf. Jede Unterrichtsidee geht unabhängig von der Aktivität/dem Thema auf einen Aspekt ein:

- Die Verlinkung der Aktivität mit dem Alltag der Lernenden und ihrer (digitalen) Umwelt.
- Die Verlinkung der Aktivität mit Konzepten aus dem Bereich der Informatik.
- Die Verlinkung der Aktivität mit Kompetenzen des Lehrplans, zunächst vorrangig aus der Domäne der Mathematik.

Die Unterrichtsideen sind auf eine Unterrichtsstunde ausgelegt. Ihre Verlinkung zu verschiedenen Bereichen und Themen erlaubt eine umfassende transversale Bearbeitung und Umsetzung.