

Computational Thinking im Unterricht

Gerald Futschek

Der Begriff Computational Thinking, im Deutschen Informatisches Denken, wurde ursprünglich von Seymour Papert in seinem für den Konstruktivismus bahnbrechenden Buch *Mindstorms: Children, computers and powerful ideas* [1] verwendet und von Jeannette Wing mit ihrem viel beachteten Artikel in der wissenschaftlichen Zeitschrift *Communications of the ACM* [2] so eindrucksvoll propagiert, dass er heute in aller Munde ist. Beide Autoren verwenden den Begriff in Zusammenhang mit jugendlichen Lernenden, obwohl Computational Thinking die Denkweisen der professionellen Computerwissenschaftler zur Lösung ihrer Problemstellungen bezeichnet. Es erhebt sich die Frage, was das Computational Thinking ausmacht, dass es für Schüler und Schülerinnen so bedeutsam ist, was davon im Unterricht verwendet werden soll und vor allem wie es unterrichtet werden kann.

Die Bedeutung von Computational Thinking

Computer und Informatik sind die aktuellen treibenden Kräfte von Wissenschaft, Industrie und Wirtschaft. Neue Erkenntnisse in den Wissenschaften, sei es Physik, Chemie, Biologie, Astronomie, Mathematik und oft auch in Geistes- und Humanwissenschaften sind ohne intelligenten Einsatz der Informationstechnologie mit ausgetüftelten Algorithmen nicht mehr denkbar. Das zeigt die Bedeutung des Computational Thinkings außerhalb der Informatik.

Für die Schule und das Erlernen durch Jugendliche müssen die Konzepte der Computerwissenschaft mit Hilfe geeigneter Beispiele und Aufgaben altersgerecht aufbereitet werden. Dies ist für alle Altersstufen möglich, altersgerechte Systeme, Roboter und Software, können das unterstützen.

Computational Thinking ist dem Mathematischen Denken in manchen Aspekten sehr ähnlich. Während das Mathematische Denken mehr auf das Beweisen von Zusammenhängen abzielt, geht es beim Computational Thinking mehr um das effiziente Erzielen von Ergebnissen. Um zu zeigen, dass die Modelle und Lösungsvorschläge des Computational Thinking richtig sind, können sie letztendlich auch am Computer ausgeführt und damit überprüft werden. Die Ergebnisse des Informatischen Denkens werden so zum Leben erweckt.

Welche Aspekte sollen unterrichtet werden?

Bei der Auswahl der Aspekte, die unterrichtet werden, kommt es darauf an, welchen Wert der Aspekt für die Allgemeinbildung und auch für das Berufsleben hat. Die amerikanische Informatiklehrer Organisation CSTA hat für die operationale Umsetzung im Unterricht die folgenden 6 Aspekte des Computational Thinking herausgearbeitet [3]:

- Formulieren von Problemstellungen in einer Weise, dass sie mit Hilfe von Computern gelöst werden können.
- Organisieren und analysieren von Daten
- Repräsentieren von Daten durch Abstraktionen wie Modelle und Simulationen
- Automatisieren von Lösungen durch algorithmisches Denken
- Finden, analysieren und implementieren von möglichen Lösungen mit dem Ziel eines sparsamen Einsatzes von Ressourcen
- Verallgemeinern und anwenden dieser Problemlösungsprozesse auf verschiedene andere Problemstellungen

Man kann deutlich erkennen, dass bei dieser Charakterisierung des Computational Thinkings der Problemlösungsprozess im Vordergrund steht. Allein die genannten Verben formulieren, organisieren, analysieren, repräsentieren, automatisieren, finden, implementieren, verallgemeinern und anwenden charakterisieren die Tätigkeiten, die relevant für das Computational Thinking sind. Es

beginnt mit dem Formulieren der Problemstellung, das die Bedeutung des sprachlichen Aspekts des Computational Thinkings hervorhebt und schließt mit dem Verallgemeinern, das zeigt, dass versucht werden soll nicht nur sehr spezielle Problemstellungen zu lösen, sondern dass das Lösungsprinzip auch auf andere ähnliche Probleme angewandt werden soll.

Alle diese Aspekte des Computational Thinkings können in unserer von Informationsverarbeitung geprägten Gesellschaft von allgemeinbildendem Wert sein. Es kommt insbesondere auf die Art der Aufgabenstellungen und Art der Vermittlung im Unterricht an, damit die Schülerinnen und Schüler die Denkweisen verinnerlichen und davon profitieren können.

Natürlich ist Informatisches Denken auch für den berufsbildenden Bereich äußerst relevant. Dort hat man den Vorteil, dass man praxisrelevante Problemstellungen bei der Hand hat.

Wie soll unterrichtet werden?

Das Denken kann nur geschult werden, wenn immer wieder neue Problemstellungen gelöst werden. Es kommt also beim Computational Thinking insbesondere darauf an, dass in ausreichendem Maße altersgerechte, dem Lernfortschritt entsprechende Aufgabenstellungen vorliegen. Die vorrangigste Aufgabe der Lehrkräfte ist es also geeignete Aufgabenstellungen zu stellen. Am besten wird das Denken durch selbständiges Lösen dieser Aufgabenstellungen geschult, erst in zweiter Linie kann durch Nachvollziehen einer vorgezeigten Lösung, gelernt werden.

Projektorientierter Unterricht und Teamarbeit sind adäquate Methoden um umfangreichere Problemstellungen des Computational Thinkings zu bearbeiten.

Die CSTA betont auch, dass Computational Thinking folgende Kompetenzen fördert:

- Vertrautheit im Umgang mit Komplexität
- Beständigkeit im Bearbeiten von schwierigen Problemen
- Toleranz für Mehrdeutigkeit
- Fähigkeit offene Fragestellungen zu behandeln
- Fähigkeit zum Kommunizieren und im gemeinschaftlichen Problemlösen

Das sind Bildungswerte, die heute immer mehr an Bedeutung gewinnen. Die Problemstellungen können gemäß dieser Kompetenzen durchaus auch offen, mehrdeutig, komplex und so schwierig sein, dass sie nur im Team und durch gute Kommunikation (sprachliche Dimension!) gelöst werden können.

Es muss auch unbedingt betont werden, dass das Computational Thinking in allen Altersstufen vom Kindergarten bis zur Reifeprüfung und natürlich auch darüber hinaus gelernt und gelehrt werden kann.

Beispiele für mustergültige Aufgabenstellungen findet man zum Beispiel bei der Computer Science Unplugged Initiative [4], bei der in vielen Aktivitäten das Informatische Denken spielerisch und ohne Verwendung von Computern erlernt wird. Bemerkenswert ist dabei, dass mit diesem Ansatz anspruchsvolle Themen der Informatik selbst Volksschulkindern vermittelt werden können.

Die Biber der Informatik Challenge [5] ist eine weitere internationale Initiative, die für alle Altersstufen eine sehr große Zahl an kurzen Aufgabenstellungen zum Computational Thinking erstellt hat. Alljährlich nehmen an dieser Challenge etwa 1 Million Schülerinnen und Schüler weltweit teil. Die Aufgaben sind so gestaltet, dass sie sowohl Schülerinnen als auch Schüler ansprechen und für das Informatische Denken begeistern können, auch oder gerade weil einige dieser Aufgaben schwieriger zu lösen sind.

Die Computerprogrammierung nimmt einen wesentlichen Teil des Informatischen Denkens ein. Es ist in unserer von Informationstechnologie und Computer geprägten Zeit notwendig, dass unsere Jugend ein klares Bild von Funktionsweise, Möglichkeiten und Auswirkungen der Informationstechnologie erhält. Nur ein Verständnis der Grundlagen der Programmierung ermöglicht hier ein klareres Bild. Darüber hinaus ist die Fähigkeit des Programmierens eine echte Ermächtigung, die es erlaubt Technologie zu bestimmen, statt von Technologie bestimmt zu werden. Computerprogrammierung kann heutzutage mit entsprechenden Systemen altersgerecht in allen Schulstufen vom Kindergarten an unterrichtet werden.

Für den Unterricht an Schulen bedeutet es, dass alle Schülerinnen und Schüler vom Kindergarten an Computational Thinking üben sollen. Sowohl ein eigenes Fach als auch die Integration in alle Fächer ist notwendig. Alle Lehrerinnen und Lehrer müssen in Computational Thinking ausgebildet bzw. fortgebildet werden.

Zusammenfassend kann man sagen, dass das Informatische Denken für alle Lebensbereiche unserer Schüler und Schülerinnen von immer größer werdender Bedeutung ist. Die Inhalte, Bildungswerte und Denkweisen sind von langfristigem Nutzen und können in verschiedensten Situationen auch unabhängig vom Computer angewandt werden.

Referenzen

- [1] Seymour Papert: *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
- [2] Jeannette Wing: *Computational thinking*. Communications of the ACM, 49(3), 33-35, 2006.
- [3] CSTA Computational Thinking Task Force:
<https://csta.acm.org/Curriculum/sub/CompThinking.html>
- [4] Computer Science Unplugged: csunplugged.org.
- [5] Biber der Informatik: www.ocg.at/biber, international: www.bebras.org.