

Das Informatikcurriculum der Hector Kinderakademien

Tsarava, K.*, Kunz, K.* und Trautwein, U.

DOI: 10.18420/ibis-02-02-06

Zusammenfassung

In den Bildungsplänen für Grundschulen der Bundesländer bleibt Informatik und informatisches Denken immer noch außen vor. Dies wird jedoch teilweise durch außerschulische Angebote kompensiert. In Baden-Württemberg und Hessen erhalten beispielsweise Grundschulkin- der die Möglichkeit, an Kursen der Hector Kinderakademien teilzunehmen. In diesem Artikel möchten wir daher die Kurse des Hector Informatikcurriculums Lehrkräften und Fachleuten im Bereich der Informatikbildung vorstellen. Das Informatikcurriculum der Hector Kinderakademien besteht aus drei Kursen und richtet sich an begabte und hochbegabte Kinder der ersten bis vierten Klassenstufe, die an Informatik interessiert sind. Bei der Entwicklung und Etablierung der Kurse wird ein evidenzbasierter Ansatz verfolgt, das heißt, dass die Kurse von der wissenschaftlichen Begleitung der Hector Kinderakademien empirisch hinsichtlich ihrer Förderwirksamkeit untersucht werden. Erste Studien belegen die Förderwirkung der Kurse.

Einleitung

Obwohl heute oft schon die Kleinsten ein eigenes Tablet haben oder mit dem Smartphone spielen dürfen, beginnt der formale Unterricht in Informatik und die Förderung des informatischen Denkens in Deutschland relativ spät, wenn man bedenkt, dass informatisches Denken (engl. *computational thinking*) als Teil der „21st century skills“ betrachtet wird. Tatsächlich hat informatisches Denken mittlerweile weltweit Einzug in die Bildungspläne gehalten, vermehrt auch in Grundschulen. In Europa waren Estland und Großbritannien 2012 Vorreiter, und seitdem haben 15 EU-Länder informatisches Denken in ihren Bildungsplänen für Grundschulen verankert (Bocconi et al., 2022). Obwohl auch die Gesellschaft für Informatik, die größte deutschsprachige Organisation für Informatik, bereits 2019 Empfehlungen für Informatikunterricht und die Förderung von informatischem Denken an Grundschulen veröffentlicht hat (Best et al., 2019), haben in Deutschland immer noch längst nicht alle Bundesländer Informatik

oder informatisches Denken in ihren Bildungsplänen für den Primarbereich aufgenommen. Dies wird teilweise mit fehlenden Lehrkräften begründet, teilweise aber auch mit dem Verweis, dass die Grundschulen zunächst die Grundlagen in Kernbereichen wie Lesen, Schreiben und Mathematik legen müssten und bereits dies bei einem hohen Prozentsatz von Schülerinnen und Schülern misslinge (vgl. Stanat et al., 2022). Der Präsident des Deutschen Lehrerverbandes, Heinz-Peter Meidinger, hat als Reaktion auf Ergebnisse der IGLU-Studie 2022 sogar gefordert, Informatik grundsätzlich aus Grundschulen zu verbannen (Epp, 2023).

Allerdings bedeutet ein Verzicht auf die Förderung informatischen Denkens in der Grundschule auch, dass vielen leistungsstarken Kindern, die großes Interesse an entsprechenden Inhalten aufweisen, eine systematische frühe Förderung in diesem Bereich verwehrt bleibt. Es ist deshalb zu überlegen, ob und wie außerunterrichtliche Angebote diese Lücke schließen können. Für ein solches außerunterrichtliches Angebot haben wir als Teil der wissenschaftlichen Begleitung der Hector Kinderakademien mehrere Kurse entwickelt, die informatisches Denken an Grundschulkindern vermitteln.

Im Jahr 2010 wurde die Hector Kinderakademie (HKA; Golle, J., Herbein, E., Hasselhorn, M., & Trautwein, 2017) in Baden-Württemberg ins Leben gerufen. Bis August 2024 sind 69 lokale Standorte der Akademie¹, die in der Regel an einer oder mehreren Grundschulen angesiedelt sind, Teil der HKA. Das Ziel der HKA ist es, Enrichment-Kurse für die 10% begabtesten Grundschulinnen in Baden-Württemberg anzubieten. In Hessen befinden sich die Hector Kinderakademien derzeit im Aufbau; die ersten Hector Kinderakademien werden dort voraussichtlich 2025 mit ihrer Arbeit beginnen. Die Hector Kinderakademie wird von der Hector Stiftung II finanziert und vom Ministerium für Kultus, Jugend und Sport Baden-Württemberg sowie dem Hessischen Kultusministerium unterstützt.

Rund 23.000 Kinder nehmen jedes Jahr am Kursangebot teil. Die HKA bietet Kurse in allen akademischen Bereichen an, hat aber einen klaren Schwerpunkt für Mathematik, Informatik,

* Geteilte Erstautorenschaft der beiden erstgenannten Autoren.

¹ Vgl. die Webseite der Hector Kinderakademie: <https://hector-kinderakademie.de>.

Naturwissenschaften und Technik, den sogenannten MINT-Fächern. Die Teilnahme an den HKA-Kursen ist freiwillig und findet außerhalb der regulären Schulzeit statt, beispielsweise im Ganztagsprogramm, nach der Schule oder am Wochenende. Alle Grundschulen in Baden-Württemberg dürfen Kinder für die HKA nominieren. Die nominierten Kinder können dann jedes Semester selbst entscheiden, an welchem bzw. welchen der angebotenen Kurse in einer nahegelegenen Akademie sie teilnehmen möchten.

Wissenschaftlich begleitet wird die Hector Kinderakademie vom Hector-Institut für Empirische Bildungsforschung an der Universität Tübingen und dem Leibniz-Institut für Bildungsforschung und Bildungsinformation (DIPF) in Frankfurt. Die wissenschaftliche Begleitung koordiniert Qualifizierungsveranstaltungen, unterstützt die Öffentlichkeitsarbeit und verantwortet das Zertifikatsstudium „Begabtenförderung und Potenzialentwicklung“ sowie die akademieübergreifenden Onlinekurse auf der Moodle-Lernplattform. Darüber hinaus umfasst die wissenschaftliche Begleitung die Evaluation der Förderwirkung des Angebots sowie die Federführung bei der Entwicklung von sogenannten Hector Core Courses. Diese werden auf der Basis des aktuellen Forschungsstandes entwickelt und ihre Wirkung mithilfe eines ambitionierten Untersuchungsdesigns überprüft.

Die drei Hector Core Courses des Hector-Informatikcurriculums² (siehe Abbildung 3) richten sich an Kinder von Klassenstufe 1 bis Klassenstufe 4. Kinder der Klassenstufen 1 und 2 können den Kurs „Planeten der Informatik“ besuchen und spielerisch und unplugged verschiedene Gebiete der Informatik kennenlernen. In Klasse 3 und 4 lernen die Kinder zu „Verstehen wie Computer denken“ und erarbeiten mithilfe von lebensgroßen Spielen und blockbasiertem Programmieren Konzepte wie Variablen, Schleifen und bedingte Verzweigungen. Dieses Wissen können Kinder der Klassenstufe 4 dann „Kreativ am Computer“ einsetzen und auf die Programmiersprache Python übertragen. Die beiden Kurse, die sich an Kinder der Klasse 3 und 4 richten, werden dabei auch in einer Online-Version angeboten. Das Hector-Informatikcurriculum ist dabei besonders auf die Förderung von informatischem Denken ausgerichtet. Um dieses Ziel zu erreichen, verwenden die Kurse unplugged-Methoden sowie block- und textbasiertes Programmieren und basieren

auf den Prinzipien des spielebasierten und verkörperlichten Lernens.

Informatisches Denken als didaktischer Ansatz

Unter informatischem Denken werden zum einen die kognitiven Prozesse und Fähigkeiten verstanden, die eine wesentliche Rolle bei der Lösung informatischer Problemstellungen und bei der Entwicklung von Programmierfähigkeiten spielen (Garcia-Peñalvo, 2016). Zum anderen werden damit erlernbare Denkstrategien bezeichnet, die für das systematische Problemlösen in Disziplinen über die Informatik hinaus einsetzbar sind. Informatisches Denken hebt sich dabei von der allgemeinen Problemlösefähigkeit ab, indem algorithmische Lösungen gefunden werden, die strategisch, systematisch, abstrakt, reproduzierbar und berechenbar sind (Barr & Stephenson, 2011).

Wenn Informatisches Denken unterrichtet werden soll, heißt das also nicht, den Kindern primär oder ausschließlich die Syntax einer Programmiersprache zu zeigen oder den Aufbau von Datenbanken zu erklären. Die Kinder sollen vielmehr lernen, komplexe Probleme zu verstehen, präzise zu formulieren, systematisch zu lösen und die entwickelte Lösung daraufhin zu implementieren, zu bewerten und zu verbessern (Wing, 2006). Um Lösungsalgorithmen für Probleme zu finden, können die Kinder dabei auf informatische Konzepte wie Sequenzen, Operatoren, Variablen, Schleifen, bedingte Verzweigungen, Ereignisse und Funktionen zurückgreifen.

Bei informatischem Denken als fachdidaktischem Ansatz stehen also diese informatischen Konzepte selbst im Vordergrund und nicht etwa ihre Umsetzung in unterschiedlichen Programmiersprachen. Damit kann informatisches Denken anfänglich sogar völlig unplugged, also ohne jegliche digitale Unterstützung, eingeübt werden, bevor es später auf praktische (Programmier-)aufgaben und schlussendlich auf fächerübergreifende Kontexte übertragen wird.

Informatisches Denken (un)plugged

Für die frühe Informatikbildung werden weitgehend „unplugged“-Übungen empfohlen (z.B. Prottsman, 2014). Unplugged (= „ausgesteckt“) meint den Verzicht auf Geräte, die Strom benötigen, also insbesondere Computer oder Roboter. Beim unplugged-Ansatz wird auf greifbare Materialien wie Stifte, Spielkarten und Spielfiguren zurückgegriffen, um den Kindern den Ein-

² Vgl. die Webseite der Hector Core Courses des Hector-Informatikcurriculums: <https://uni-tuebingen.de/de/123280>.



Abbildung 1: Ein Beispiel für eine "unplugged"-Übung im Spiel "Schildkröten und Krabben".

stieg in die Informatik zu erleichtern und sie nicht mit Technologien zu überfordern.

Der unplugged-Ansatz hat dabei den Vorteil, dass nicht zuerst Programmieren gelernt werden muss, bevor tiefere Einblicke in Bereiche der Informatik möglich werden. Er implementiert zudem den didaktischen Ansatz des Konstruktivismus, indem die Kinder die Möglichkeit bekommen sich selbstständig mit kinästhetischen Übungen zu beschäftigen (Tsarava et al., 2017, 2018). Dadurch kann zum einen bereits sehr früh mit der Förderung von informatischem Denken begonnen werden, zum anderen kann eine große Bandbreite von Kindern, insbesondere auch Mädchen, für die Informatik begeistert werden, die hier sonst unterrepräsentiert sind (Bell & Vahrenhold, 2018).

Allerdings besteht bei ausschließlicher Verwendung von unplugged-Übungen die Gefahr, dass

künstliche und kontextlose Lernszenarien dominieren, weshalb die erlernten Fähigkeiten auch auf „plugged-in“-Übungen übertragen werden sollen. Eine Möglichkeit, Kinder schon früh an informatische Geräte heranzuführen, sind zum Beispiel Lernroboter wie Beebots (Seckel et al., 2023) oder Ozobots (Körper et al., 2021), die mit einfacher Programmierung von den Kindern gesteuert werden können.

Blockbasiertes, hybrides und textbasiertes Programmieren

Der Einstieg ins eigentliche Programmieren geschieht mittlerweile fast ausschließlich in blockbasierten Programmierumgebungen wie Scratch oder Code.org, bzw. mit den spezifischen Programmierumgebungen der verwendeten Lernroboter. Beim blockbasierten Programmieren werden Programmierblöcke wie Puzzleteile in- und aneinandergeschoben. Dadurch wird Programmieren möglich, ohne dass den Kindern die Tastatur vertraut sein muss oder sie die genaue Syntax der Programmiersprache kennen müssen. Die Kinder können zudem aus einem Angebot an vorgefertigten Blöcken geeignete Anweisungen auswählen und mithilfe von „Drag- und Drop“ an die gewünschten Stellen ziehen. Der Vorteil dabei ist, dass die Kinder Anweisungen nicht im Vorfeld erlernen und sich merken müssen. Als ein Nachteil von blockbasiertem Programmieren wird allerdings gesehen, dass es trotz seiner Möglichkeiten für sehr komplexe Programme von den Kindern oft nicht

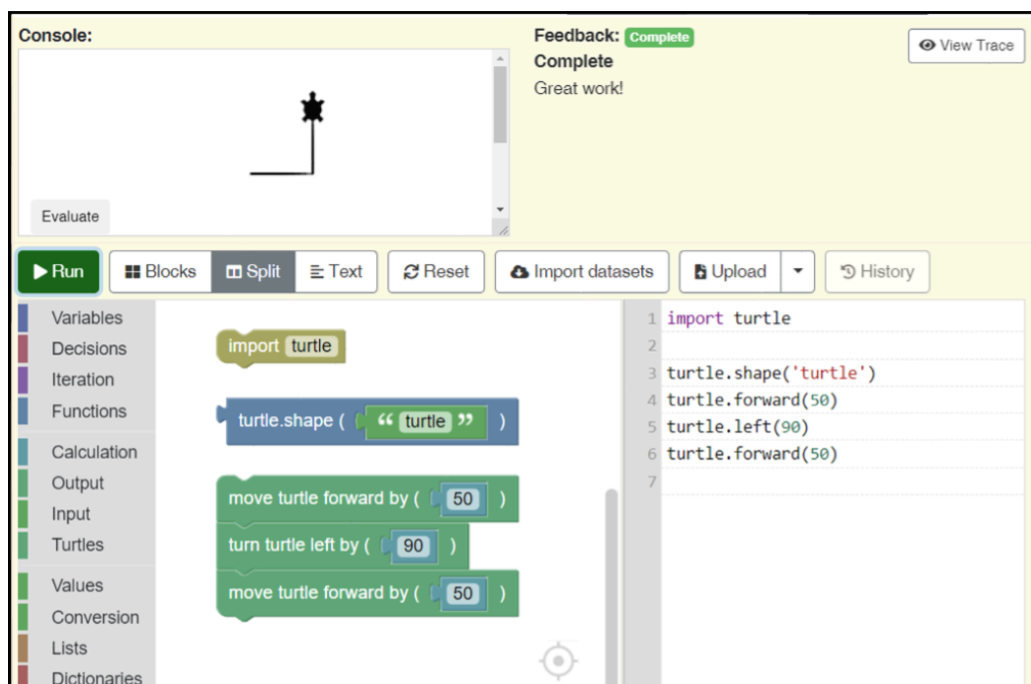


Abbildung 2: Die dual hybride Programmierumgebung Blockly mit der Möglichkeit zum parallelen block- und textbasierten Programmieren.

als „richtiges Programmieren“ angesehen wird (Weintrop & Wilensky, 2017).

Zudem hat sich gezeigt, dass blockbasiertes Programmieren einen späteren Übergang zu textbasiertem Programmieren nicht immer erleichtert. Untersuchungen geben ein gemischtes Bild, ob Kinder die erlernten Konzepte einfach übertragen können. Der Übergang kann außerdem dazu führen, dass die Motivation und positive Einstellung der Kinder fürs Programmieren sinken, wenn zusätzliche Schwierigkeiten wie Syntax, Programmiersprachen in Englisch oder auch das Tastaturschreiben dazukommen (Alrubaye et al., 2020).

Eine Möglichkeit, diesen Übergang weniger abrupt zu gestalten, sind sogenannte hybride Programmierumgebungen (z.B. Weintrop & Wilensky, 2017). Dabei kann blockbasiertes Programmieren zum einen in einer textbasierten Umgebung integriert werden, indem die Programmbearbeitung blockbasiert stattfindet und der bearbeitete Abschnitt anschließend automatisch in Text übersetzt wird. Zum anderen können blockbasiertes Programmieren und textbasiertes Programmieren dual nebeneinander stehen und Veränderungen in einer Modalität werden automatisch auf die andere übertragen.

Spielerbasiertes und verkörperlichtes Lernen

Wenn Inhalte und Kompetenzen anhand von Spielen oder spielerischen Aufgaben vermittelt werden, wird dies als spielerbasiertes Lernen bezeichnet. In der Informatik gibt es dabei erstens die Möglichkeit Aufgaben zu gamifizieren, indem mithilfe von gelösten Aufgaben Punkte

oder Levels erreicht werden können. Dieser Ansatz wird zum Beispiel mit Code.org verfolgt. Zweitens können informatische Aufgaben direkt Teil des Spiels sein und müssen als Teil der Geschichte des Spiels gelöst werden. Drittens können Kinder mit ihren Programmierkenntnissen auch schon früh eigene Spiele entwickeln, was zum Beispiel bei Scratch ermöglicht wird.

Verkörperlichtes Lernen (auf Englisch: embodied learning) basiert auf der Theorie, dass viele Aspekte des Denkens stark vom physischen Körper oder von physischen Gegenständen abhängig sind (Barsalou, 2008). Der unplugged-Ansatz, aber auch die Verwendung von Robotern basieren auf diesen Grundlagen.

Die Hector-Informatikkurse und Evidenz für ihre Wirksamkeit

Um Kinder der Hector Kinderakademien in ihrem informatischen Denken zu fördern, haben wir drei Kurse entwickelt, die die Methoden des spielerbasierten und verkörperlichten Lernen einsetzen: „Planeten der Informatik“, „Verstehen wie Computer denken“ und „Kreativ am Computer“. Zusätzlich gibt es einen Kurserweiterung, die sich „Planet des Internets“ nennt.

Das Hector-Informatikcurriculum, basierend auf den GI-Kompetenzen für Informatische Bildung, zielt darauf ab, Schüler*innen dabei zu unterstützen, Algorithmen in ihrer Lebenswelt anzuwenden, algorithmische Grundbausteine zu nutzen und Algorithmen in Alltagssprache zu beschreiben. Darüber hinaus sollen sie Algorithmen mit den algorithmischen Grundbau-

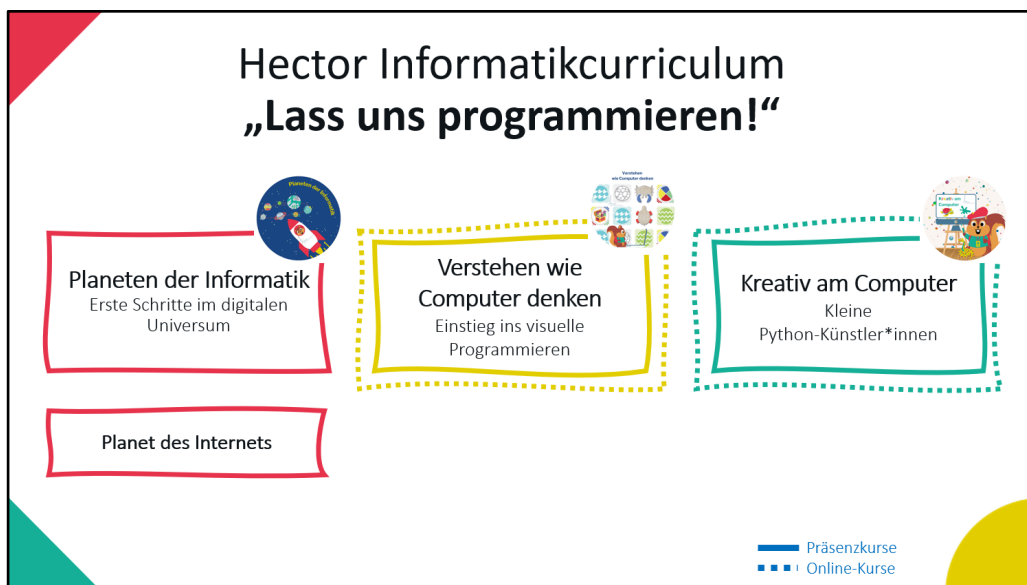


Abbildung 3: Die Kurse des Hector-Informatikcurriculums.

steinen Anweisung, Sequenz, Wiederholung und Verzweigung entwerfen, realisieren und testen, Algorithmen in verschiedenen formalen Darstellungsformen darstellen, Algorithmen unter Verwendung der Fachsprache vergleichen und ein Informatiksystem programmieren.

Das Curriculum zielt außerdem darauf ab, die Schüler*innen dabei zu fördern, Automaten in ihrer Lebenswelt als selbstständig arbeitende Maschinen zu beschreiben, Zustände von Automaten zu benennen und zu erläutern, dass ein Automat regelgesteuert seine Zustände verändert. Damit unterstützt es die Schülerinnen und Schüler auch dabei, zu verstehen, dass ihre Lebenswelt von Informatik durchdrungen ist.

Planeten der Informatik

Der Kurs „Planeten der Informatik“ lässt Kinder der ersten und zweiten Klasse in die Informatik-Galaxie eintauchen, wo sie mit den Außerirdischen, den Heckis, von Planeten zu Planeten reisen. Die Kinder lernen auf dem ersten Planeten, dem „Heimatplaneten“ der Heckis, was Befehle sind, wie sie zu Sequenzen zusammengefügt werden, wie diese Sequenzen zum Beispiel in bunten Perlenarmbändern vorkommen und dass ein Algorithmus so etwas wie Anleitung zum Kressepflanzen sein kann. Auf dem „irren Planeten“ müssen die Kinder dann Befehle und Sequenzen anwenden, um sich gegenseitig aus Irrgärten zu befreien. Auf dem „Planet Krypto“ lernen die Kinder und die Heckis den Außerirdischen „Löffel“ kennen,



Abbildung 4: Die gebastelten Heckis auf dem Automatenplaneten.

welcher nur in Löffelsprache spricht und den Kindern andere Außerirdische vorstellt, die alle in merkwürdigen Geheimsprachen wie Pixelsprache oder Blinzelsprache sprechen.

Auf dem vierten Planeten, dem „Automaten-Planet“, erfahren die Kinder wie Automaten mit Zuständen und Zustandsübergängen dargestellt werden können und haben selbst die Möglichkeit Automaten, die sie in ihrem Alltag benutzen so darzustellen. Befehle, Sequenzen und Algorithmen finden auch ihre Anwendung

auch auf dem „Planet Rhythmo“, auf dem die Kinder eigene Musik komponieren können, die dann mit dem eigenen Körper als Schlagzeug aufgeführt werden kann. Auf den Planeten sechs („tanzender Planet“) und sieben („verzweigter Planet“) lernen die Kinder kleine Roboter, die Ozobots, kennen und wie man diese mit gemalten Linien und verschiedenen Farben zum Tanzen bringt und über verzweigte Wege führen kann. Anschließend kehren die Kinder mit den Heckis zurück und erzählen den zurückgebliebenen Heckis von ihrer Reise, indem sie alle Planeten nochmal Revue passieren lassen.

Der Kurs hält sich bei der Wahl der Inhalte an die von der Gesellschaft für Informatik vorgegeben Themen und die zum Ende der zweiten Klasse zu erreichenden Kompetenzen wie „Informationen und Daten“ (Planet Krypto), „Algorithmen“ (Heimatplanet, irrer Planet, Planet Rhythmo, verzweigter Planet) und „Sprachen und Automaten“ (Automaten Planet), „Informatiksysteme“ (Automaten-Planet, tanzender bzw. verzweigter Planet). Das Themengebiet „Informatik, Mensch und Gesellschaft“ wird über alle Kurssitzungen, also Planeten, betont, indem die Kinder lernen die theoretischen Konzepte mit ihrer Umgebung und ihrem Alltag in Verbindung zu bringen, wie zum Beispiel Sequenzen in der Musik oder Automaten auf dem Schulweg. Der Kurs findet dabei bis auf die Verwendung der Ozobots unplugged statt und selbst die Ozobots werden nur zur Unterstützung des verkörperlichten Ansatzes verwendet.

Planet des Internets

Anschließend an den Kurs „Planeten der Informatik“ können Kinder der dritten und vierten Klasse den Kurs „Planet des Internets“ besuchen. In zwei Doppelstunden lernen die Kinder hier, wie das Internet als Netzwerk von Netzwerken aufgebaut ist, wie Webseiten aufgerufen werden und Nachrichten über das Internet verschickt werden können. Der Kurs verwendet dafür Spiele, die die Funktionsweise veranschaulichen. Als Einstieg wird eine Variante von „Stille Post“ gespielt. Alle Kinder bekommen eine Dobble-Karte mit sechs verschiedenen Symbolen und nur Kinder mit einem bestimmten Symbol werden als Verbindungspunkte des Senders und Empfängers der „Stillen-Post“-Nachricht eingesetzt. Das darauffolgende Netzwerk-Spiel lässt die Kinder nur mit Hilfe leitender Fragen der Kursleitung ein Netzwerk bilden, in welchem Nachrichten übermittelt werden können. Die Kinder entwickeln hier selbstständig Ideen die Adressen oder Routern entspre-

chen. Schließlich verkörperlichen die Kinder im Internet-Verbindungs-Spiel einzelne Elemente des Internets wie den Anbieter oder den Router, nachdem der Ablauf des Spiels mithilfe von

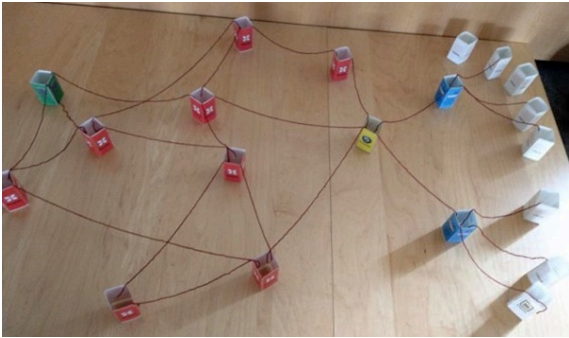


Abbildung 5: Das Internet-Vernetzungs-Spiel.

Kartonfiguren gezeigt wurde. Die Materialien dieses Kurses sind als Open Educational Resource (OER) unter folgendem Link verfügbar http://hdl.handle.net/10900.3/OER_IMOLFSPE.

Verstehen wie Computer denken

Im Gegensatz zu den Planeten-Kursen, die sich inhaltlich auch auf verschiedene Bereiche der Informatik wie Automaten, Kryptographie oder Netzwerke beziehen, fokussiert sich der Kurs „Verstehen wie Computer denken“ ganz besonders auf die Förderung von informatischem Denken.

Mit dem speziell für den Kurs entwickelten lebensgroßen Spiel „Schildkröten und Krabben“ (Tsarava, Moeller, et al., 2019) werden die Kinder spielerisch an die grundlegenden informatischen Konzepte (Sequenzen, Schleifen, bedingte Verzweigung und Ereignisse) herangeführt. „Schildkröten und Krabben“ enthält eine Serie von drei Lernspielen: „Die Schatzsuche“, „Die Muster“ und „Das Wettrennen“, welche entweder als Karten- oder Brettspiel konzipiert sind. Die Spiele wurden bewusst nicht als digitale Spiele entwickelt, um die Kinder erleben zu lassen, dass informatische Konzepte nicht nur in digitalen Umgebungen anwendbar sind. Sie zielen darauf ab kognitive Prozesse des informati-

schen Denkens wie algorithmisches Denken, Abstraktion und Mustererkennung zu trainieren. Die Brettspiele sind als OER unter folgendem Link verfügbar http://hdl.handle.net/10900.3/OER_MDCKSMXP.

Der Kurs geht schrittweise dazu über die im Spiel erlernten Konzepte und Prozesse in der Programmierumgebung Scratch³ und der „open-hardware“-Plattform Arduino einzusetzen. Im letzten Modul des Kurses verwenden die Kinder Open Roberta Lab⁴ eine interaktive Robotersimulation, um mit einer einfach erlernten blockbasierten Programmiersprache eigenständig Probleme zu lösen. Das Kursmanual für diesen Kurs ist zu finden bei Leifheit (2020, S. 118-367).

Kreativ am Computer

Kreativ am Computer ist der letzte Kurs des Informatikcurriculums und richtet sich an Kinder der vierten Klasse, die schon etwas Programmiererfahrung haben und den Übergang zu textbasiertem Programmieren machen wollen. Dafür wird die dual hybride Programmierumgebung BlockPy (für weitere Informationen siehe Bart et al., 2017) verwendet, die den Kindern jederzeit erlaubt, sowohl block- wie auch textbasiert zu programmieren. BlockPy basiert auf der Programmiersprache Python und erlaubt die Verwendung des Python Moduls Turtle Graphics⁵, welches die Steuerung einer simulierten Schildkröte ermöglicht. Die Schildkröte hält dabei einen Stift, der durch die Bewegung über den Bildschirm bunte Linien zeichnen kann.

Der Kurs folgt vom didaktischen Ansatz dem EIS-Prinzip (Bruner, 2009), bei dem zuerst mithilfe von Metaphern ein bestimmtes Konzept enaktiv (unplugged) eingeführt wird, im nächsten Schritt die Kinder das Konzept mithilfe der

³ Scratch <https://scratch.mit.edu/>.

⁴ Open Roberta Lab <https://lab.open-roberta.org/>.

⁵ Für die Dokumentation des Moduls siehe <https://docs.python.org/3/library/turtle.html>.

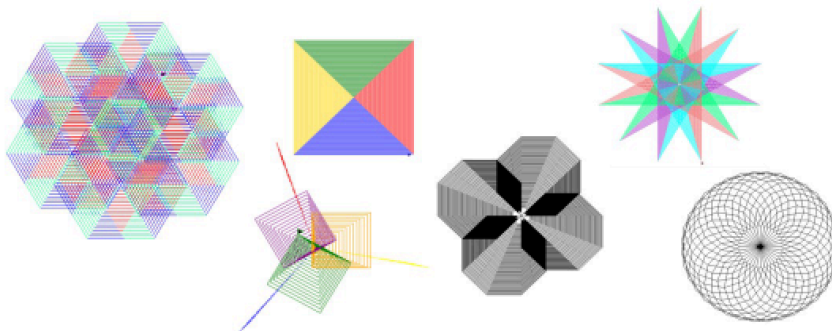


Abbildung 6: Beispiele für die von den Kindern programmierten Bilder im Kurs „Kreativ am Computer“.

Programmblöcke ikonisch verwenden und schlussendlich beim textbasierten Programmieren symbolisch einsetzen. Die Metaphern sind dabei zum Beispiel eine Box für das Konzept der Variable. Die Box kann mit einem Namen beschriftet werden (Variablenname) und dieser Name steht als Platzhalter für den Inhalt (Variablenwert). Eine Variablenzuweisung entspricht also dem Hinzufügen von einem Zettel mit einer darauf gedruckten Zahl in die Box. Zu jedem so eingeführten Konzept folgen zwei Aufgaben für die Kinder. Zuerst sollen die Kinder eine Aufgabe lösen, die einen bestimmten Lösungsweg und einen geplanten Einsatz des neuen Konzepts verlangt, und dann folgt eine Aufgabe, bei der die Kinder das Konzept kreativ einsetzen und eigene Bilder erstellen können. Zum Abschluss des Kurses programmieren die Kinder ein Computerspiel. Die erste Online-Version des Kurses ist als OER unter folgendem Link verfügbar http://hdl.handle.net/10900.3/OER_UEVLLJBZ.

Die Wirksamkeit der Kurse

Die evidenzorientierte Strategie (siehe Abbildung 7) der Hector Kinderakademien sieht vor, alle Kurse auf ihre Wirksamkeit zu überprüfen. Nachdem spezifische Bedürfnisse und Lernziele identifiziert wurden, werden auf der Basis aktueller wissenschaftlicher Erkenntnisse das Kurskonzept und die didaktische Umsetzung entwickelt. Anschließend durchlaufen alle Kurse mehrere Stufen der empirischen Überprüfung: Zuerst wird der Kurs im kleinen Rahmen (ca. 30 Kinder) von den Kursentwicklerinnen auf die Durchführbarkeit untersucht (Pilotierung). Dabei werden Materialien und die geplanten Zeiten für einzelne Übungen angepasst und eine erste Einschätzung getroffen, ob der Kurs seine Ziele erfüllt. Beim Informatikcurriculum beinhalten die Ziele natürlich zuvorderst die Förderung informatischen Denkens. Es werden aber auch die Auswirkungen des Kurses auf Motivation, Selbstkonzept und Kreativität der Kinder untersucht.

In nächsten Schritt („Wirksamkeitsstudie“; rund 100 Kinder) wird die Wirksamkeit des Kurses auf die Erreichung der Ziele untersucht, wenn gezielt ausgewählte Kursleitungen an den Hector Kinderakademien den Kurs unterrichten und sich dabei strikt an den Vorgaben des Kurshandbuchs orientieren. In einem weiteren Schritt („Effektivitätsstudie“; rund 200 Kinder) wird untersucht, ob der Kurs seine Wirksamkeit beibehält, wenn der Kurs von regulären Kursleitungen der jeweiligen Akademien unterrichtet wird, die zuvor eine Fortbildung erhalten haben. In einem letzten Schritt kann („Scaling-up“) geprüft werden, inwieweit die Wirkung des Kurses auch nach vollständiger Implementation im Programm stabil ist.

Der Kurs „Verstehen wie Computer denken“ hat die drei Phasen von Pilotstudie bis Effektivitätsstudie durchlaufen. Wir konnten dabei feststellen, dass er einen signifikant positiven Effekt hatte und sich das informatische Denken der Kinder verbessert hatte (Tsarava, 2020; Tsarava, Leifheit, et al., 2019). Auch die Motivation und das Selbstkonzept für das Programmieren der Kinder, die am Kurs teilnahmen, war im Vergleich zu Kindern der Kontrollgruppe signifikant höher (Leifheit, 2020; Leifheit et al., 2019).

Zum jetzigen Zeitpunkt haben die Kurse „Planeten der Informatik“, „Planet des Internets“ und „Kreativ am Computer“ nur die Pilotierung durchlaufen. Die Praxistauglichkeit aller drei Kurse konnte dabei festgestellt werden. Die Pilotstudien signalisiert, dass der Kurs „Planet des Internets“ mittelstarke bis starke Effekte auf die Korrektur von Missverständnissen von Grundschulkindern über das Internet hat (Nutz et al., 2024). Ebenso deuten die Pilotstudien darauf hin, dass der Kurs „Kreativ am Computer“ die informatischen Denkfähigkeiten der Kinder verbessert und potenziell den Übergang von blockbasiertem zu textbasiertem Programmieren erleichtern kann, während die Motivation der Kinder erhalten bleibt (Kunz et al., 2023).

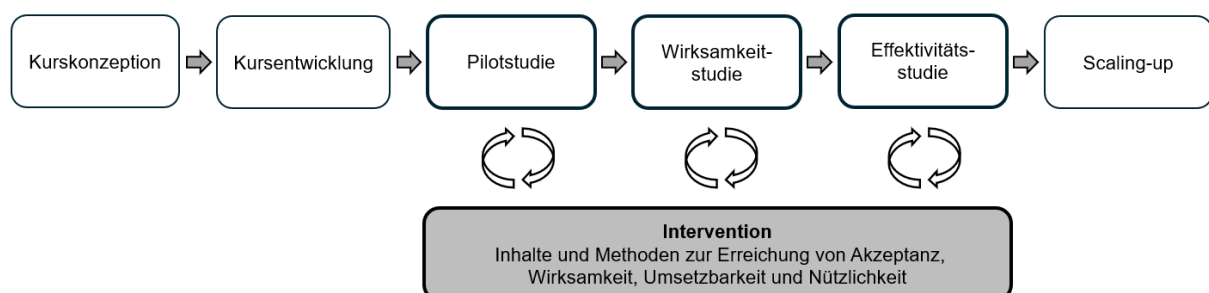


Abbildung 7: Phasen der Kursentwicklung und -bewertung als Interventionen (Abbildung adaptiert von Trautwein et al., 2023).

Fazit

Das Hector-Informatikcurriculum der Hector Kinderakademien zielt darauf ab, eine Lücke in der informatischen Bildung von begabten und hochbegabten Kindern zu schließen, die an ihren Grundschulen keinen Informatikunterricht erhalten. Die von uns durchgeführten empirischen Feldversuche haben gezeigt, dass diese Kinder nicht nur Interesse an informatischen Konzepten haben und diese früh verstehen können, sondern auch, dass informatisches Denken auf angemessene Weise früh gefördert werden kann. Dies ist eine wichtige Erkenntnis, da informatisches Denken als Problemlösungskompetenz über die Bereiche der Informatik hinaus eingesetzt werden kann.

Unsere eigenen Arbeiten zum informatischen Denken von Grundschulkindern sowie viele weitere außerunterrichtliche und außerschulische Angebote können natürlich nicht direkt die Frage beantworten, inwieweit Informatik und informatisches Denken grundsätzlich in die Bildungspläne der Länder für die Grundschule aufgenommen werden sollten. Sie verdeutlichen aber, dass die Kultusministerien sich ernsthaft mit diesem Thema auseinandersetzen sollten.

Soll informatisches Denken in der Grundschule unterrichtet werden, werden dafür ausgebildete Lehrkräfte benötigt, die auf aktuelle, altersgerechte Unterrichtsmaterialien zugreifen können. Ohne dies wird das Vorwissen der Kinder in der Sekundarschule aufgrund der Nutzung außerunterrichtlicher Angebote sehr heterogen sein, was einen noch differenzierteren und adaptiveren Unterricht erfordert, damit alle Schüler*innen entsprechend ihres Lernstands gefördert werden. Aufgrund der Ergebnisse unserer Arbeiten und der Bedeutung von informatischem Denken als „21st-century skill“ plädieren wir für die frühe Implementierung informatischer Inhalte und deren Aufnahme in die Bildungs-Lernpläne für Grundschulen.

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 08.08.2024.

Alrubaye, H., Ludi, S., & Mkaouer, M. W. (2020). Comparison of block-based and hybrid-based environments in transferring programming skills to text-based environments. *CASCON 2019 Proceedings - Conference of the Centre for Advanced Studies on Collaborative Research - Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, April, 100–109.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>

Barsalou, L. W. (2008). Grounded Cognition. *Annual Review of Psychology*, 59(August), 617–645. <https://doi.org/10.1146/annurev.psych.59.103006.093639>

Bart, A. C., Tibau, J., Tilevich, E., Shaffer, C. A., & Kafura, D. (2017). BlockPy: An Open Access Data-Science Environment for Introductory Programmers. *Computer*, 50(5), 18–26. <https://doi.org/10.1109/MC.2017.132>

Bell, T., & Vahrenhold, J. (2018). CS Unplugged—How Is It Used, and Does It Work? (pp. 497–521). https://doi.org/10.1007/978-3-319-98355-4_29

Best, A., Borowski, C., Büttner, K., Freudenberg, R., Fricke, M., Haselmeier, K., Herper, H., Hinz, V., Humbert, L., Müller, D., Schwill, A., & Thomas, M. (2019). Kompetenzen für informatische Bildung im Primarbereich. *Gesellschaft für Informatik e.V.*

Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). State of play and practices from computing education REVIEWING COMPUTATIONAL THINKING IN COMPULSORY EDUCATION. <https://doi.org/https://dx.doi.org/10.2760/126955>

Bruner, J. S. (2009). The process of education. In Bruner, Jerome S. *The process of education*. Harvard university press, 2009.

Epp, E. (2023, May 19). Präsident des Lehrerverbandes fordert: Grundschule soll sich auf drei Fächer konzentrieren. *Stern*. <https://www.stern.de/gesellschaft/grundschule--lehrerverband-praesident-will-zwei-faecher-streichen-33482516.html>

Garcia-Peñalvo, F. J. (2016). What Computational Thinking Is. *Journal of Information Technology Research*, 9(3), v–vi(October).

Golle, J., Herbein, E., Hasselhorn, M., & Trautwein, U. (2017). Talentförderung in der Grundschule durch Enrichment: Das Beispiel der Hector-Kinderakademien. In U. Trautwein & M. Hasselhorn (Ed.), *Begabungen und Talente* (Hogrefe, pp. 191–210).

Körber, N., Bailey, L., Greifenstein, L., Fraser, G., Sabitzer, B., & Rottenhofer, M. (2021). An Experience of Introducing Primary School Children to Programming using Ozobots (Practical Report). *The 16th Workshop in Primary and Secondary Computing Education*, 1–6. <https://doi.org/10.1145/3481312.3481347>

Kunz, K., Moeller, K., Ninaus, M., Trautwein, U., & Tsarava, K. (2023). Making the Transition to Text-

Based Programming: The Pilot Evaluation of a Computational Thinking Intervention for Primary School Students. Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research, 1–6. <https://doi.org/10.1145/3605468.3609770>

Leifheit, L. (2020). The Role of Self-Concept and Motivation Within the “ Computational Thinking ” Approach to Early Computer Science Education. <http://dx.doi.org/10.15496/publikation-55314>

Leifheit, L., Tsarava, K., Moeller, K., Ostermann, K., Golle, J., Trautwein, U., & Ninaus, M. (2019). Development of a Questionnaire on Self-concept, Motivational Beliefs, and Attitude Towards Programming. Proceedings of the 14th Workshop in Primary and Secondary Computing Education, 1–9. <https://doi.org/10.1145/3361721.3361730>

Nutz, M., Kunz, K., & Tsarava, K. (2024). Development and Empirical Assessment of an Intervention on the Internet’s Structure and Functioning for Third and Fourth Graders. 2024 IEEE Global Engineering Education Conference (EDUCON), 01–10. <https://doi.org/10.1109/EDUCON60312.2024.10578725>

Prottzman, K. (2014). Computer science for the elementary classroom. *ACM Inroads*, 5(4), 60–63. <https://doi.org/10.1145/2684721.2684735>

Seckel, M. J., Salinas, C., Font, V., & Sala-Sebastià, G. (2023). Guidelines to develop computational thinking using the Bee-bot robot from the literature. *Education and Information Technologies*, 28(12), 16127–16151. <https://doi.org/10.1007/s10639-023-11843-0>

Stanat, P., Schipolowski, S., Schneider, R., Sachse, K. A., Weirich, S., & Henschel Sofie (Eds.). (2022). IQB-Bildungstrend 2021: Kompetenzen in den Fächern Deutsch und Mathematik am Ende der 4. Jahrgangsstufe im dritten Ländervergleich. Waxmann Verlag.

Trautwein, U., Golle, J., Jaggy, A., Hasselhorn, M., & Nagengast, B. (2023). Mutual benefits for research and practice: Randomized controlled trials in the Hector Children’s Academy Program. *Annals of the New York Academy of Sciences*, 1530(1), 96–104. <https://doi.org/10.1111/nyas.15074>

Tsarava, K. (2020). Computational Thinking as a Cognitive Construct: Cognitive Correlates, Assessment & Curriculum Design [University of Tübingen]. <http://dx.doi.org/10.15496/publikation-55320>

Tsarava, K., Leifheit, L., Ninaus, M., Román-González, M., Butz, M. V., Golle, J., Trautwein, U., & Moeller, K. (2019, October 23). Cognitive correlates of computational thinking: Evaluation of a blended unplugged/Plugged-in course. *ACM International*

Conference Proceeding Series. <https://doi.org/10.1145/3361721.3361729>

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training Computational Thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, 5(2), 25–44. <https://doi.org/10.17083/ijsg.v5i2.248>

Tsarava, K., Moeller, K., & Ninaus, M. (2019). Board games for training computational thinking. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: Vol. 11385 LNCS. https://doi.org/10.1007/978-3-030-11548-7_9

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M. V., Trautwein, U., & Ninaus, M. (2017). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. Proceedings of the 11th European Conference on Games Based Learning, ECGBL 2017, October, 687–695.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education*, 18(1). <https://doi.org/10.1145/3089799>

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Dr. Katerina Tsarava
E-Mail: katerina.tsarava@uni-tuebingen.de

Katrin Kunz
E-Mail: katrin.kunz@uni-tuebingen.de

Prof. Dr. Ulrich Trautwein
E-Mail: ulrich.trautwein@uni-tuebingen.de

Hector-Institut für Empirische Bildungsforschung