

Rieseneder, Michael; Wagner, Wolfgang Erstes Programmieren mit Kindern über Handlungserfahrungen. Das Konzept Activity-based-Coding

Holub, Barbara [Hrsg.]; Himpl-Gutermann, Klaus [Hrsg.]; Mittlböck, Katharina [Hrsg.]; Musilek-Hofer, Monika [Hrsg.]; Varelija-Gerber, Andrea [Hrsg.]; Grünberger, Nina [Hrsg.]: *lern.medien.werk.statt. Hochschullernwerkstätten in der Digitalität. Bad Heilbrunn : Verlag Julius Klinkhardt 2021, S. 215-230. - (Lernen und Studieren in Lernwerkstätten)*



Quellenangabe/ Reference:

Rieseneder, Michael; Wagner, Wolfgang: Erstes Programmieren mit Kindern über Handlungserfahrungen. Das Konzept Activity-based-Coding - In: Holub, Barbara [Hrsg.]; Himpl-Gutermann, Klaus [Hrsg.]; Mittlböck, Katharina [Hrsg.]; Musilek-Hofer, Monika [Hrsg.]; Varelija-Gerber, Andrea [Hrsg.]; Grünberger, Nina [Hrsg.]: *lern.medien.werk.statt. Hochschullernwerkstätten in der Digitalität. Bad Heilbrunn : Verlag Julius Klinkhardt 2021, S. 215-230* - URN: urn:nbn:de:0111-pedocs-228220 - DOI: 10.25656/01:22822

<https://nbn-resolving.org/urn:nbn:de:0111-pedocs-228220>

<https://doi.org/10.25656/01:22822>

in Kooperation mit / in cooperation with:



<http://www.klinkhardt.de>

Nutzungsbedingungen

Dieses Dokument steht unter folgender Creative Commons-Lizenz: <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.de> - Sie dürfen das Werk bzw. den Inhalt unter folgenden Bedingungen vervielfältigen, verbreiten und öffentlich zugänglich machen sowie Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen: Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen. Dieses Werk bzw. der Inhalt darf nicht für kommerzielle Zwecke verwendet werden. Die neu entstandenen Werke bzw. Inhalte dürfen nur unter Verwendung von Lizenzbedingungen weitergegeben werden, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind. Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use

This document is published under following Creative Commons-License: <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en> - You may copy, distribute and transmit, adapt or exhibit the work in the public and alter, transform or change this work as long as you attribute the work in the manner specified by the author or licensor. You are not allowed to make commercial use of the work. If you alter, transform, or change this work in any way, you may distribute the resulting work only under this or a comparable license.

By using this particular document, you accept the above-stated conditions of use.



Kontakt / Contact:

peDOCS
DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation
Informationszentrum (IZ) Bildung
E-Mail: pedocs@dipf.de
Internet: www.pedocs.de

Mitglied der:


Leibniz-Gemeinschaft

Lernen und Studieren in Lernwerkstätten



Barbara Holub / Klaus Himpsl-Gutermann
Katharina Mittlböck / Monika Musilek-Hofer
Andrea Varelija-Gerber / Nina Grünberger
(Hrsg.)

lern.medien.werk.statt

Hochschullernwerkstätten in der Digitalität

Barbara Holub
Klaus Himpsl-Gutermann
Katharina Mittlböck
Monika Musilek-Hofer
Andrea Varelija-Gerber
Nina Grünberger
(Hrsg.)

lern.medien.werk.statt

Hochschullernwerkstätten in der Digitalität

Verlag Julius Klinkhardt
Bad Heilbrunn • 2021

k

Dieser Titel wurde in das Programm des Verlages mittels eines Peer-Review-Verfahrens aufgenommen. Für weitere Informationen siehe www.klinkhardt.de.

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation
in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten
sind im Internet abrufbar über <http://dnb.d-nb.de>.

2021.ig. © by Julius Klinkhardt.

Coverfoto: © ZLI PH Wien.

Druck und Bindung: AZ Druck und Datentechnik, Kempten.

Printed in Germany 2021.

Gedruckt auf chlorfrei gebleichtem alterungsbeständigem Papier.



Die Publikation (mit Ausnahme aller Fotos, Grafiken und Abbildungen) ist veröffentlicht unter der Creative Commons-Lizenz: CC BY-NC-SA 4.0 International
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

ISBN 978-3-7815-5904-2 digital

doi.org/10.35468/5904

ISBN 978-3-7815-2468-2 print

Inhaltsverzeichnis

<i>Barbara Holub, Klaus Himpf-Gutermann, Katharina Mittlböck, Monika Musilek-Hofer, Andrea Varelija-Gerber und Nina Grünberger</i>	
Einführung in den Band	11

Zur Konstituierung von Hochschullernwerkstätten

<i>Sandra Tänzer</i>	
Mut – Zumutung – Ermutigung. Oder wie man lernt, eine Hochschullernwerkstatt zu lieben	23

<i>Markus Peschel, Hartmut Wedekind, Pascal Kihm und Mareike Kelkel</i>	
Hochschullernwerkstätten und Lernwerkstätten – Verortung in didaktischen Diskursen	40

<i>Dietlinde Rumpf und Corinna Schmude</i>	
Von der Herausforderung, die Vielfalt von Hochschullernwerkstätten in einer Definition abzubilden	53

<i>Pascal Kihm und Markus Peschel</i>	
„Komplexität wagen!“ – Methoden zur Beforschung von offenen Lehr-Lern-Prozessen in Hochschullernwerkstätten	70

Zum Spannungsverhältnis von Hochschullernwerkstätten und Digitalität

<i>Mark Weißhaupt, Ralf Schneider, Clemens Griesel und Agnes Pfrang</i>	
Digitale Erfahrung? Über das Lernen zwischen Instruktion und (Ko-)Konstruktion	87

<i>Stefan Brée, Lena S. Kaiser und Tanja Wittenberg</i>	
Lernwerkstatt als digitaler Erfahrungsort. Potenziale und Herausforderungen für Lernwerkstätten als Orte „offener multimedialer Produktionsästhetik“	103

<i>Ulrike Stadler-Altmann und Gerda Winkler</i>	
Real & virtuell, analog & digital: Dimensionen einer Kooperation. Multifunktionalität als Kennzeichen zukunftsfähiger Lernwerkstatt- und Bibliothekskonzeption	121

Josef Buchner und Michael Kerres

Lernwerkstattarbeit in der digital vernetzten Welt.

Die Perspektive der gestaltungsorientierten Mediendidaktik 137

Sabrina Tietjen und Silvia Thünemann

Forschungswerkstatt digital: ein modernes Lehr-Lernarrangement

für eine Digitalisierungsstrategie im Lehramt? 147

Zu fach- und mediendidaktischen Perspektiven

Johannes Mayer, Antonia Lemensieck, Maria Reinhardt und Karl Wollmann

Fachliche Perspektiven auf digitalisierungsbezogene Lernangebote

in der Ausbildung von Grundschullehrer*innen 163

Heike Hagelgans

Die Thematisierung digitaler Medien in den schulpraktischen Studien

Fachdidaktische Reflexionen von digitalen Medien für das Lernen im

Mathematikunterricht der Primarstufe 179

Jeanette Hoffmann, Katharina Egerer und Franziska Herrmann

Analoge Bilder – digitaler Film.

Möglichkeitsräume für literarästhetisches Lernen in

Hochschullernwerkstätten 196

Michael Rieseneder und Wolfgang Wagner

Erstes Programmieren mit Kindern über Handlungserfahrungen.

Das Konzept Activity-based-Coding 215

Patrick Isele und Julia Höke

Reflexionen über digitales Lernen mit dem Sphero SPRK+

Erfahrungen mit Studierenden in der „Lernwerkstatt“³⁴ 231

Susanne Schumacher, Ulrike Stadler-Altmann und Enrico A. Emili

Piktogramme als Unterstützungsmedien.

Studien zu Effekten von Bilderbüchern mit zusätzlichen Symbolen

im Kindergarten 248

Zu Aspekten, Perspektiven und Herausforderungen von Hochschullernwerkstätten

Sabine Fischer und Max de Baey-Ernsten

Didaktische Perspektiven einer ästhetischen Werkstatt 273

Tanja Wittenberg und Lena S. Kaiser

„Ich war frustriert, ich bin kein Kind mehr“ – Erfahrungslernprozesse mit verwendungs- und bedeutungsoffenem Material in kindheitspädagogischen Hochschullernwerkstätten 291

Lisa Eßel und Laura Schlichting

Hochschullernwerkstätten im Spannungsfeld zwischen Freiwilligkeit und Pflicht 309

Mareike Kelkel, Markus Peschel und Pascal Kihm

Potenziale der pädagogisch-didaktischen Öffnung in Hochschullernwerkstätten 321

Barbara Holub und Sybille Roszner

Hochschullernwerkstatt – Ausgangspunkt für persönlichkeitsorientierte Professionalisierung in der Ausbildung für Lehrpersonen 334

Verzeichnis der Autor*innen 348

Michael Rieseneder und Wolfgang Wagner

Erstes Programmieren mit Kindern über Handlungserfahrungen

Das Konzept Activity-based-Coding

Informatik ist der Schlüssel zum Verständnis der Informationstechnik und damit der Schlüssel zum Verstehen unserer modernen Umwelt! (Gallenbacher, 2017, S. VIII)

Abstract

Activity-based-Coding ist ein im Education Innovation Studio an der Pädagogischen Hochschule Oberösterreich entworfenes didaktisches Konzept, mit dem bei Kindern aus dem Primarstufenbereich über konstruktivistisches bzw. konstruktionistisches Vorgehen Grundvorstellungen zu informatischen und mathematischen Begriffen generiert werden können, die für das Programmieren bzw. Algorithmisieren benötigt werden. Die Vorgehensweise bietet eine natürliche Differenzierung an, bei der Kinder selbst ihre Vorgehensweise in Bezug auf Schwierigkeit, Tempo und Abstraktionsgrad wählen können. Zentraler Punkt dieses Konzepts ist das E-I-S-Prinzip nach Bruner und damit verbunden eine möglichst enge Passung der Intermodalitätswechsel der Repräsentationsebenen (enaktiv, ikonisch, symbolisch). Im Sinne des Spiralprinzips nach Bruner werden von einer ersten Bewusstmachung von Algorithmisierungen, stets darauf aufbauend, weitere Kompetenzen für Computational Thinking entwickelt. Dabei führt der didaktische Weg von Bewegungsspielen über den Einsatz von konkreten Bodenrobotern hin zu virtuellen Programmierungsumgebungen. Die Methoden und Medien werden so gewählt, dass über das eigene Tun abstrakte informatische Ideen derart entwickelt werden, dass den Abstraktionen stets eigene Handlungen innewohnen. Activity-based-Coding wird an der PH Oberösterreich seit dem Schuljahr 2017/18 entwickelt und erforscht.

1 Einleitung

Die Auswirkungen des Leitmedienwechsels sind vergleichbar mit denjenigen bei der Erfindung und Verbreitung des Buchdrucks und betreffen alle Bereiche unseres Lebens. Honegger, 2017, S. 31)

Die Digitalisierung unserer Gesellschaft ist in vollem Gange und löst zunehmend einen Leitmedienwechsel vom gedruckten Buch aus. Nahezu jeder Bereich unseres täglichen Lebens ist davon betroffen. Vorteile wie Nutzungen des Internets, des E-Mailverkehrs, der Navigation über Online-Karten, den Möglichkeiten von Smartphones, etc. sowie mögliche Gefahren wie dem Verlust von Arbeitsplätzen, dem Verlust der Privatsphäre, dem Verkauf persönlicher Daten u. v. m. seien hier nur beispielhaft erwähnt. Ein kompetenter Umgang mit digitalen Medien ist daher notwendig (Zorn, Trappe, Stöckelmayr, Kohn, & Derndorfer, 2013) und bedarf dreier Kompetenzen (Honegger, 2017, S. 77): einer Anwendungskompetenz, einer Kompetenz, mit Medien kritisch umzugehen, und einer informatischen Kompetenz im Sinne des von Jeanette Wing (2006) geprägten Begriffs des Computational Thinking.

Dieser Beitrag stellt eine mögliche didaktische Umsetzung einer Entwicklung informatischen Denkens (Activity-based-Coding) für Kinder der Primarstufe vor, die in erster Linie auf der konstruktivistischen Lerntheorie vor allem nach Piaget, dem sich daraus ableitendem E-I-S Prinzip von Bruner und dem Konzept der Grundvorstellungen nach von Hofe basiert. Nach dem Spiralprinzip wird in Übungsformen bestehendes Wissen verfeinert und erweitert und darauf bauend können neue Begrifflichkeiten eingeführt werden.

Activity-Based-Coding stellt eine Möglichkeit zum Generieren von Kompetenzen des Computational Thinking und hier insbesondere des Programmierens bzw. Algorithmisierens bei Kindern der Primarstufe dar. Bei der Begrifflichkeit wird auf Aho (2012) zurückgegriffen, wonach Computational Thinking sämtliche angewandte Denkprozesse beim Formulieren von Problemen beinhaltet, deren Lösungen in Form von Algorithmen ausgedrückt werden können. Wenngleich derzeit die Informatik im österreichischen Lehrplan für die Volksschule keinen eigenen Bereich abdeckt, lassen sich darin doch zahlreiche informatische Konzepte wie beispielsweise das Arbeiten mit Algorithmen finden (Sabitzer, Antonitsch & Passterk, 2014). Activity-based-Coding stützt sich überwiegend auf Prinzipien der Mathematikdidaktik. Die mathematisch-didaktischen Prinzipien können – Informatik ist historisch aus der Mathematik heraus entstanden – auf die informatische Didaktik übertragen werden kann (Savard & Highfield, 2015).

Daraus ableitend wird der Einsatz derzeit zugänglicher Medien vorgestellt, der sowohl möglichst passende Übergänge innerhalb der Repräsentationsstufen ermöglicht als auch im Sinne des Spiralprinzips zunehmend informatisches Wis-

sen generieren kann. Übergeordnetes Ziel ist die Ausbildung der Fähigkeit, digitale Medien selbst gestalten und bestimmen zu können, um nicht von digitalen Medien bestimmt zu werden. Ein solches Generieren von informatischen Grunderfahrungen ist bereits im frühen Kindesalter möglich (Schwill, 2001).

2 Das Konzept Activity-based-Coding

Wenn Kinder einen Programmierbefehl aus einer eigenen Handlung heraus verstehen lernen, dann lernen sie nicht nur den Befehl, sondern Prinzipien des Programmierens und was ein Programm eigentlich ist. (Wagner)

Seit dem Schuljahr 2017 wird das Konzept Activity-based-Coding an der Pädagogischen Hochschule Oberösterreich entwickelt und beforscht. Ziel ist das Generieren von informatischen Grunderfahrungen vor allem bei Kindern der Primarstufe.

Auf Grundlage des E-I-S Prinzips nach Bruner (1971) – also den Repräsentationen von mathematischen Begriffen auf den Ebenen der Handlung (enaktiv), des Bildes (ikonisch) und der Abstraktion in Form von Symbolen und Sprache – sollen informatische Begriffe mit Sinn erfüllt werden. Diese Sinnkonstituierung soll durch möglichst enge Verzahnungen der Repräsentationsebenen untereinander generiert werden. Dadurch können im Sinne des Spiralprinzips neu zu lernende Inhalte entwickelt werden, die stets mit bereits bekanntem Wissen in Verbindung gebracht werden. Grundvorstellungen (Vom Hofe, 1995; Wartha, 2010) können durch das Gelingen vom Wechsel der Repräsentationsebenen zu einer Begrifflichkeit angebahnt werden. Die einer Abstraktion innewohnenden, inhärenten Strukturen (Aebli, 2011, S. 207ff.) können somit durch beabsichtigte Handlungen ausgedrückt werden und gleichfalls strukturierte Handlungen in abstrakte Zeichen oder Sprache übertragen werden. Ebenso soll eine Reversibilität der informatischen Begriffe (z. B. left turn) von der Abstraktion zur Handlung (bzw. der bildhaften Beschreibung einer Handlung) und umgekehrt auch in deren Bewegungsausführung (left turn = 3 x right turn) ermöglicht werden. Das Konzept Activity-based-Coding geht bei der Vermittlung informatischer Begriffe so weit als möglich von Repräsentationen auf der Ebene der Handlung aus, die in der Folge mit den Ebenen des Bildes und jener der Abstraktion verbunden werden. Diese Verbindungen sind dann gelungen, wenn ein mentales Modell generiert werden kann (Rabel, 2011). Ziel ist, dass in der Abstraktion das Wissen innewohnt, welches auf den Ebenen der Handlung und des Bildes generiert wurde. Im Sinne eines konstruktivistischen (Piaget & Szeminska, 1975) bzw. konstruktivistischen (Papert & Harel, 1991) Vorgehens sollen die Kinder dazu animiert

werden, sich auf spielerische Art und Weise eigene Umgebungen zur Begriffsbildung zu gestalten und in diesen Umgebungen zu handeln. Dieses Handeln erfolgt im Sinne einer natürlichen Differenzierung (Krauthausen, Scherer, & Scherer, 2017), wobei die Kinder in ihrem eigenen Tempo in von ihnen gewählten Komplexitätsstufen arbeiten können. Die Auseinandersetzung mit den Lerninhalten soll mit anderen in Partner*innen- bzw. in Gruppenarbeit erfolgen, wodurch ein Austausch und eine Erweiterung von Gedankengängen sowie eine Korrektur von Fehlvorstellungen ermöglicht wird. Ebenso können durch adäquate Aufgabenstellungen die prozessbezogenen Kompetenzen im Sinne des Problemlösens, des Argumentierens, des Modellierens und des Operierens gefördert werden. Activity-based-Coding übernimmt bei vielen handlungsbezogenen informatischen Aufgabenstellungen Prinzipien von Bell, Alexander, Freeman und Grimley (2009) im Sinne von „Computer Science Unplugged“.

Mögliche Verbindungen der informatischen Inhalte mit der Lebenswelt der Kinder setzt eine Verbindung zu sämtlichen Gegenständen der Primarstufe voraus, wie dies im österreichischen Lehrplan der Volksschule (BMBWF, 2020, S. 9ff.) gefordert wird.

3 Mögliche praktische Umsetzungsform

Im Folgenden wird anhand des Konzepts Activity-based-Coding eine praktische Vorgehensweise zur Generierung von Computational Thinking bei Kindern der Primarstufe vorgestellt.

Jedwede mathematische Operation ist eine Handlung – egal ob arithmetisch oder geometrisch – wenn sie im Bewusstsein ihrer inhärenten Beziehungen ausgeführt wird (Piaget & Szeminska, 1975). Daher ist die zentrale Ausgangslage der vorgestellten Möglichkeiten zu informatischen Begriffsbildungen das eigene Handeln, das eigene Tun, das mit den Strukturen der Begriffe inhärent ist. Darauf aufbauend können im Sinne des Spiralprinzips zunehmend Abstrahierungen erfolgen.

3.1 Von der Generierung algorithmischen Denkens zu ersten Programmiererfahrungen

Ein Algorithmus ist eine präzise Beschreibung von Aktivität. Algorithmen sind die Grundlage jedes Computerprogramms und – allgemein – jedes Systems mit vorhersehbarem Verhalten. (Weigend, 2009)

Algorithmen sind Teil des Alltags der Kinder, werden jedoch kaum als solche benannt. Spielregeln erklären, nach Plan bauen, tanzen etc. seien hier nur auszugswise genannt. Der Zugang zur Bewusstmachung von Algorithmen kann

über das Roboterspiel als Bewegungsspiel mit Kindern oder Spielfiguren verwirklicht werden.

3.1.1 Kinder spielen selbst Roboter

Als eine mögliche Form der Sinnkonstituierung von Algorithmen wird das Roboterspiel „Kommando Pimperle“ gewählt. Bei diesem Bewegungsspiel führt ein Kind als *Ingenieur*in* ein anderes Kind, das die Rolle eines *Roboters* einnimmt, zu einem vereinbarten Ziel. Die Bewegungsformen des Roboters entsprechen den Ausführungen von (jeweils einem Schritt) *vorwärts* und *rückwärts* und den Drehbewegungen im rechten Winkel (bzw. Viertelkreis) nach links und rechts in den Benennungen *Rechtsdrehung* und *Linksdrehung*. Die genannten Handlungen entsprechen in ihrer Anweisung Programmbefehlen, wie sie in der von Papert (1980) entwickelten Programmiersprache LOGO angeführt wurden. Die genannten Befehle sind Basis für in Folge erwähnte Medien wie Bee-bots, Cubetto, etc. und weiteren Programmiersprachen wie Ozoblockly, Scratch und Swift. Die Befehle, die im Roboterspiel verwendet werden, werden somit im Sinne des Spiralprinzips nach Bruner weiterverwendet und -entwickelt.

Der geplante Weg kann mit ca. schrittgroßen Kartonpfeilen und Drehkärtchen ausgelegt werden, wodurch dieser auf Bildebene dargestellt wird (siehe Abb. 1). Die Kartonpfeile sowie die Drehkärtchen entsprechen in ihrer Gestaltung der Notation der Befehle in symbolhafter Form.

Das Ingenieurskind und das Roboterkind sowie ein mögliches beobachtendes Kind wechseln bei jedem Durchgang der Steuerung zu einem vereinbarten Ziel ihre Rollen. Die Befehle werden anfangs sprachlich ausgedrückt und in weiterer Folge dem Roboterkind zusätzlich oder ausschließlich in symbolhafter Form mitgeteilt. Die Auswahl der Notation der Symbole in Form von Pfeilen oder in den Abkürzungen FD (forward – vorwärts), BK (back – rückwärts), RT (right turn – Rechtsdrehung) und LT (left turn – Linksdrehung) erfolgt im Sinne einer natürlichen Differenzierung von den Kindern selbst. Die Notation der Abfolge der Befehle entspricht einem Programm. Das Programm ist automatisiert, da (bei gleicher Schrittlänge) vom selben Ausgangspunkt immer derselbe Zielpunkt erreicht wird. Auch im Sinne einer natürlichen Differenzierung kann zuerst die Bewegung erfolgen und im Anschluss daraus die Abstraktion in Form eines Befehls bzw. eines Programms oder in Form eines Schätzens zuerst ein Programm bzw. ein Programmabschnitt erstellt werden, das in Folge über die Handlung überprüft wird.

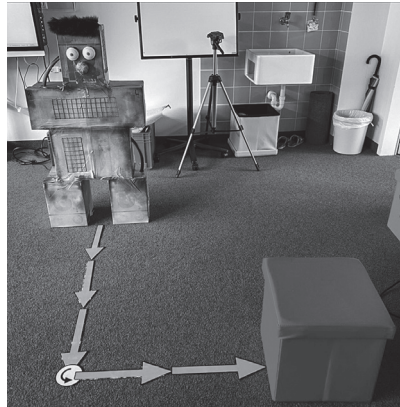


Abb. 1: Kartonroboter mit ausgelegten Befehlskarten

Als Problemlösungsaufgabe kann etwa dienen, dass der Roboter ein Quadrat bzw. ein Rechteck abschreiten soll. Im Sinne einer dynamischen Geometrie kann beim Roboterspiel vom eigenen Handeln ein Intermodalitätswechsel auf die Symbol Ebene in Form eines Programmes stattfinden. Das Programm beschreibt den Umfang der Figur mit Befehlen. 2FD, RT, 2FD, RT, 2FD, RT, 2FD, RT zeigt eine mögliche Notation eines Programms, das den Umfang eines Quadrates mit der Seitenlänge von 2 Schritten definiert. Dabei kann eine Iteration von Befehlen anhand der Vervielfachung der Bewegung 2FD, RT (statt FD, FD, RT) erkannt werden. Aus dem gesamten Programm (2FD, RT, 2FD, RT, 2FD, RT, 2FD, RT) kann die Iteration 4-mal (2FD, RT) erkannt werden. Durch den Ausschluss eines Befehls, z. B. RT, kann über den Problemlösungsprozess, dass 3 LT zum selben Ziel führt, ein Verständnis der Reversibilität der Bewegungen generiert werden.

Mögliche natürliche Differenzierungen:

- freie Auswahl der Start- und Zielpunkte
- freie Auswahl der Strecken (also der Lösungen)
- Ingenieurskind geht hinter dem Roboterkind (Positionswechsel)
- Ingenieurskind bleibt auf seiner Position stehen (Embodiment)
- Roboterkind geht die Abfolge der Befehle in einem Zug, abschnittsweise oder Schritt für Schritt
- die Handlung kann dem Befehl oder der Befehl der Handlung folgen (Reversibilität der Abfolge von Handlung und symbolhafter Notation beim Programmieren)

Mögliches generiertes Wissen:

- Algorithmisierungen von Bewegungsabläufen
- Modellierung von Bewegungsabläufen zu einem versprachlichten/notierten Programm

- Embodiment: Ingenieurskind kann sich mental in das Roboterkind hineinversetzen
- Entwicklung von Grundvorstellungen zu den Befehlen bzw. dem Konzept von Befehlen basierend auf eigenem Tun; Vertiefen durch ständige Intermodalitätswechsel
- Entwicklung von Grundvorstellungen zu einem Programmbegriff auf Handlungsebene
- Iterationen bei Vervielfachen von Befehlen
- Korrigieren falscher Lösungen (Debuggen)
- Problemlösen bei der Erstellung eines Programmes zu vorgegebenen Zielpunkten
- Argumentieren und Begründen bei der Findung von Lösungen
- Unterbrechung eines Bewegungsablaufes (Pause) beim gemeinsamen Agieren mehrerer Kinder zur Kollisionsvermeidung

3.1.2 Transformation des Roboterspiels mit Kindern hin zu Spielfiguren

Nachdem die Kinder als „handelnde Roboter“ Handlungserfahrungen zu Programmbefehlen gesammelt haben, kann das Roboterspiel auf einem ca. DIN-A3-großen Spielraster miniaturisiert werden. Auf dem Spielraster können genauso Befehlskärtchen ausgelegt und eine Spielfigur zu einem Ziel geführt werden. Die Vorbereitungen sowie der Platzbedarf für dieses Spiel sind im Vergleich zum vorher beschriebenen Roboterspiel weitaus geringer. Das Auslegen der Programme gelingt den Kindern schneller. Zudem stellt dieses Spiel einen Repräsentationswechsel bei der Handlung vom eigenen Tun zum Führen einer Spielfigur hin dar. Sämtliche natürliche Differenzierungen können wie beim Roboterspiel (3.1.1) ebenso durchgeführt werden. Die Wissensgenerierung der in 3.1.1 beschriebenen Aspekte können, wenn schon vorhanden, verstärkt werden, bzw. wenn nicht vorhanden, neu entwickelt werden.

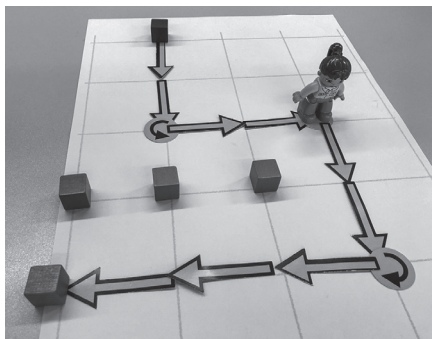


Abb. 2: Miniaturisierung des Roboterspiels mittels Spielfiguren, die auf einem Spielraster geführt werden

3.2 Transformation der bisherigen Programmiererfahrungen auf Bodenroboter

Erfahrungen zu den Programmbefehlen Vorwärtsbewegung, Rückwärtsbewegung, Linksdrehung und Rechtsdrehung wurden bisher auf sämtlichen Repräsentationsstufen gemacht, mit dem Ziel (nie endende) Grundvorstellungen zu entwickeln. Als Bodenroboter, der in seinem Befehlsumfang im Wesentlichen auf diese Befehle beschränkt ist, wird der Bee-bot gewählt. Die Darstellung der Befehlstasten des Bee-bot ähnelt den ursprünglich verwendeten Befehlskärtchen bzw. der symbolhaften Notation. Dem Befehl Pause ist bei dem Bee-bot eine extra Taste zugewiesen. Die Ausführung von Pausen zur Verhinderung von Kollisionen ist den Kindern bereits vom gemeinsamen Bewegen her bekannt.

Ein Anwenden der Programmierkenntnisse auf einen Bodenroboter stellt hinsichtlich der Arbeit in den jeweiligen Repräsentationsstufen eine Änderung dar: Die Befehle müssen durch Druck auf die Befehlstaste gegeben werden; dies kann der Symbolebene zugeordnet werden. Die Darstellung der Befehlstasten ist symbolhaft vom Roboterspiel her bekannt und hat in ihrer Darstellung noch bildhafte Elemente. Ein schrittweises Bewegen des Bee-bot ist auf der Handlungsebene möglich und kann zur Modellierung eines möglichen Lösungsweges durchgeführt werden. Sehr häufig wird von den Kindern die Steuerung des Bee-bots über die Tasten gewählt, sodass dieser sich selbst bewegt. Die Steuerung einer Maschine, die selbst gewählte Befehle durchführt, stellt für Kinder eine Motivation dar.

Die Reversibilität der Abfolge von Handlung und symbolhafter Darstellung ist nicht mehr gegeben. Der Bee-bot muss zuerst programmiert werden, um sein Programm ausführen zu können. Zur Dokumentation der Programme müssen die Befehle von den Kindern selbst notiert werden, da der Bee-bot dazu keine Darstellungsform bietet. Tippfehler beim Programmieren können nicht korrigiert werden und erfordern das Löschen und die Neueingabe des gesamten Programms. Dies gilt ebenso für Programmierfehler, was bei aufwändigen Programmen zu Motivationsverlust führen kann.

3.3 Verständnisgenerierung von Funktionen

Eine kindgerechte Darstellung von Funktionen bietet beispielsweise der Bodenroboter Cubetto an. Dieser wird über ein Steuerbrett, auf dem Befehle mittels Plastikplättchen der Reihe nach aufgelegt werden können, programmiert. Über eine Steuerungstaste wird das Programm kabellos auf den Bodenroboter Cubetto übertragen, der dieses sodann ausführt. Bei der Ausführung des Bodenroboters blinkt auf der Programmierzeile unter dem jeweils aktiven Befehl ein kleines blaues Licht, wodurch das Handeln des Roboters und das abstrakte Programm verbunden werden. Dies ermöglicht eine Kontrolle und ein allenfalls notwendiges Debuggen. Dadurch wird eine zusätzliche simultane Verbindung von Symbole-

bene und Bildebene erreicht, die mental mit den bereits generierten Grundvorstellungen verzahnt werden kann.

Zum Kennenlernen des Cubetto müssen die bereits bekannten mit Handlung verbundenen Darstellungen von Befehlen auf die dem Cubetto eigenen Befehlsdarstellungen in Form von Plättchen abstrahiert werden. Befehle müssen vom Beginn der Programmierzeile lückenlos gelegt werden. Funktionen können als Variable beschrieben werden, die einen oder mehrere Befehle umfasst. Funktionen werden über einen blauen Baustein abgerufen, der separat von der Programmdarstellung liegt (siehe Abb. 3).

Kinder können im Sinne eines eigenständigen und forschenden Lernens die Bedeutung der Befehlsfunktion durch Ausprobieren selbsttätig entwickeln. Die Beschreibung eines Kindes für eine Funktion bzw. deren Aufruf über Cubetto lautete: „Das (blaue Plättchen für eine Funktion) macht das, was da unten (Funktionsleiste) ist.“ Über die Aufgabenstellung, den Roboter den Umfang eines Quadrates abschreiten zu lassen, kann durch Kürzen des Programms 2 FD, LT, 2 FD, LT, 2 FD, LT, 2 FD, LT mittels der Funktion 2FD, LT und dessen vierfachen Abruf eine mögliche erste Anwendung im Sinne des Problemlösens stattfinden. Dabei kann auf die bereits aus dem Roboterspiel bzw. vom Bee-bot bekannten Aspekte der Quadratprogrammierung zurückgegriffen werden.

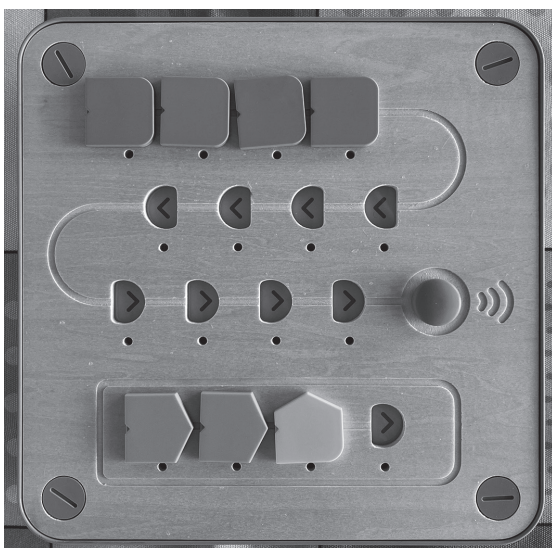


Abb. 3: Steuerbrett des Cubetto mit der Programmierung eines Quadrates über Funktionen

Abbildung 3 zeigt unten die Funktionsdarstellung von 2 FD, LT, die oben in der Programmierzeile des Hauptprogramms über vier Funktionsbefehle aufgerufen

wird. Der Bodenroboter führt somit den Bewegungsablauf des Umfangs eines Quadrates mit der Seitenlänge 2 Einheitslängen aus.

3.4 Verständnisgenerierung von Schleifen

Der Bodenroboter Matatalab bietet die Möglichkeit des Einsatzes von Schleifen bei der Notation von Programmen an. Wie der Bodenroboter Cubetto wird Matatalab über ein Steuerbrett mit Befehlsplättchen programmiert. Über eine Aktionstaste führt Matatalab das mit Befehlsplättchen gelegte Programm aus. Die bereits bekannten Bewegungsbefehle vorwärts und rückwärts können durch Vielfachungsplättchen iteriert werden. Funktionen können in einer eigenen Zeile zuerst als solche definiert und danach programmiert und im Hauptprogramm durch Auflegen des Befehlsplättchens *fn* aufgerufen werden. Durch das mittels Cubetto generierte Wissen über den Funktionsbegriff kann eine Abstrahierung hin zu den rein abstrakten Befehlsplättchen von Matatalab erfolgen.

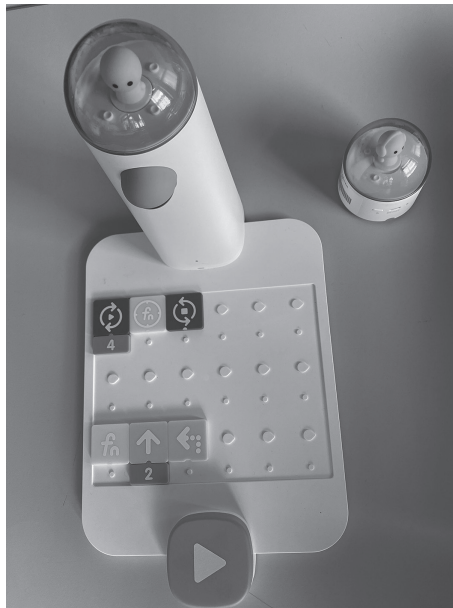


Abb. 4: Matatalab: Programmierung eines Quadrates mit Funktion und Schleife

Der bereits aus dem Roboterspiel bekannte Umstand, dass sich bei einem Quadrat Befehlsfolgen wiederholen, kann für das Verständnis von Schleifen genutzt werden, wie ein Kind der 1. Schulstufe es ausdrückte: „Da machst du immer das gleiche – also viermal.“ In Abb. 4 ist die Programmierung eines Quadrates mittels Schleife notiert, in der sich die Funktion *fn* befindet. Die Schleife wird

viermal wiederholt. Die Funktion ist getrennt vom Hauptprogramm über das Befehlsplättchen *fn* ausgewiesen. Diese Form der Programmierung zeigt ein mögliches Endprodukt einer Programmierung mit Schleifen und Funktionen. Es ist zu empfehlen, das Verständnis der Iterationsbefehle, des Funktionsbegriffs und von Schleifen nacheinander zu generieren.

3.5 Generierung eines Verständnisses für blockbasiertes Programmieren

Die bisher vorgestellten Programmierweisen entsprechen einer Programmierung mit aufeinanderfolgenden einzelnen Befehlsschritten, wobei jeder Befehl durch eine einzige Ausführung determiniert ist. Bei der blockbasierten Programmierung, bei der Block für Block programmiert wird, können Blöcke mehrere Ausführungen des Befehls beinhalten. Im Sinne des Spiralprinzips sollen ausgehend von eigenen Handlungen Grundvorstellungen aufgebaut werden, die ein Verständnis für diese Art des Programmierens ermöglichen. Die Darstellung eines Programmes in Form von Blöcken entspricht einer gängigen Form (Struktogramme).

3.5.1 OSMO

OSMO basiert auf dem RAIT-Prinzip (Reflective – Artificial – Intelligence – Technology), bei dem über einen Spiegel die Kamera eines iPads physische Elemente erkennen kann, die vor dem Gerät liegen (Broda & Frank, 2015). Bei dem dazugehörenden Spiel Coding Awbie kann eine virtuelle Figur (Awbie) mittels physischer Programmierblöcke gesteuert werden. Auf der Handlungsebene können Parameter der Programmierblöcke manipuliert werden. Die Bewegungsblöcke (Abb. 5) sind in deren Richtungs determination durch Drehen des Richtungspfeiles auf der Handlungsebene manipulierbar. Zahlplättchen, die den Programmierblöcken angehängt werden, ermöglichen deren Wiederholung. Sämtliche Plättchen sind magnetisch und haften den Programmierschriften gemäß aneinander.



Abb. 5: konkrete Programmblöcke OSMO

OSMO soll Gelegenheit zu ersten Erfahrungen mit blockbasiertem Programmieren bieten. Die Handlungen finden durch Verändern der Programmierblöcke statt. Das zu steuernde Objekt ist – im Gegensatz zu den konkreten Bodenrobotern – eine virtuelle Figur in einer virtuellen Umgebung.

3.5.2 Blockbasiertes Programmieren in Verbindung mit dem Roboterspiel

Durch OSMO ist das Prinzip der Aneinanderkettung von Programmierblöcken bekannt und kann nun auf weitere Programmierungsumgebungen übertragen werden. Als Programmierungsumgebung, die sich mit der Steuerung konkreter Roboter verbinden lässt, kann die Programmiersprache Ozoblockly herangezogen werden. Mit den in Abb. 6 dargestellten Programmierblöcken können auf Handlungsebene Styrodurblöcke (Eigenbau) in deren Abfolge sowie in deren Steuerung durch Parameter (wie fast – slow; left – right; etc.) bestimmt werden. Die Befehle zur Bewegung sind durch move fix vorgegeben und können durch die Parameter forward oder backward bzw. die fixen Parameter speed und distance in verschiedenen Geschwindigkeits- und Entfernungsparametern determiniert werden. Ziel der Steuerung kann ein Roboterkind sein, das wie unter 3.1.1 beim Roboterspiel zu einem Ziel gesteuert wird. Sämtliche unter 3.1.1 angeführten Lernprozesse sind somit auf eine blockbasierte Programmierweise übertragbar. Dadurch wer-

den die bereits bekannten Kenntnisse vertieft und können im Sinne des Spiralprinzips weiterentwickelt werden.

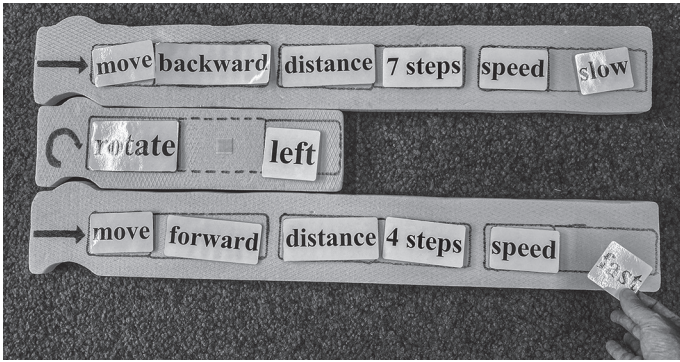


Abb. 6: Eigenbau Styrodurblöcke mit Befehlen und Parameterkärtchen

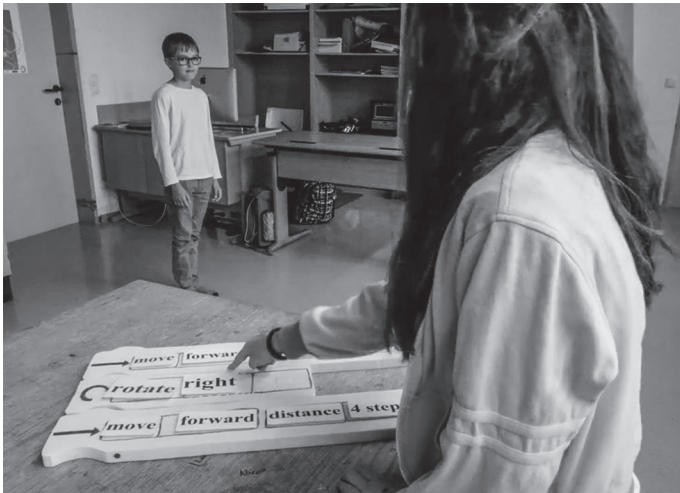


Abb. 7: Roboterspiel mit Programmblöcken

3.5.3 Blockbasiertes Programmieren mit Bodenrobotern

Die im Roboterspiel durch Steuerung mittels Programmierblöcken aus Styrodur gemachten Erfahrungen können auf die Programmierungsumgebung Ozoblockly übertragen werden. In der virtuellen Programmierungsumgebung Ozoblockly (online oder App) werden die konkreten Bodenroboter Ozobots programmiert und gesteuert. Über die Steuerung ist eine Überprüfung der Programmierung aufgrund der Bewegungsausführungen und Tonausgaben der Ozobots möglich.

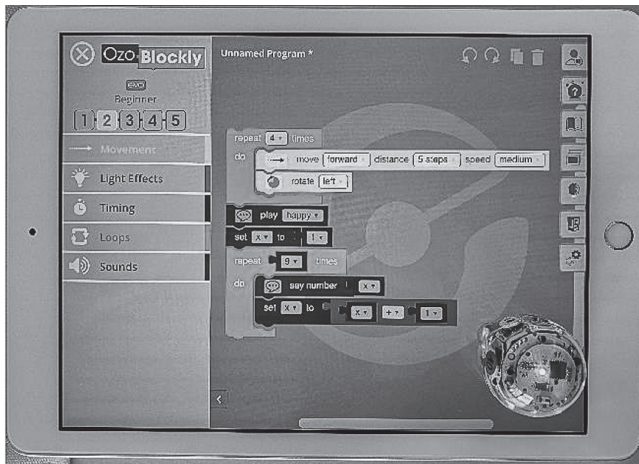


Abb. 8: Programmierungsumgebung Ozoblockly mit Bodenroboter Ozobot

Die Anwendungsmöglichkeiten der Bodenroboter Ozobot können sowohl für die Primarstufe genutzt als auch auf Aufgabenstellungen für die Sekundarstufe ausgeweitet werden (Geier & Ebner, 2017). Ozobots bieten Möglichkeiten für Aufgabenstellungen mit natürlicher Differenzierung vom Primarstufenalter bis zur Tertiärstufe, beispielsweise bei der gemeinsamen Arbeit von Volksschulkindern mit Studierenden.

3.6 Einsatz erster textbasierter Programmiersprachen

Um Erfahrungen mit rein textbasierten Programmiersprachen zu sammeln, kann Swift Playgrounds herangezogen werden. Bei Swift Playgrounds wird mit virtuellen Textbausteinen (Programmierbefehlen) eine virtuelle Figur in ihren Bewegungsausführungen gesteuert. Bei der Notation der Programme kann aus vorgegebenen Textbausteinen ausgewählt werden, wodurch mögliche Eingabefehler bei der Notation von Befehlen verhindert werden.

Eine möglichst enge Verbindung zur Handlungsebene kann durch Nachbauen von Swift-Playgrounds-Umgebungen erfolgen (siehe Abb. 9). Dabei können in dieser Umgebung die Kinder selbst (oder in miniaturisierter Form Spielfiguren) mittels selbst angefertigten Textkärtchen aus Papier mit den jeweiligen aufgedruckten Programmierbefehlen zu einem Ziel gesteuert werden, was wiederum der Vorgangsweise eines Roboterspiels entspricht. Die Umgebung kann somit gleichzeitig sowohl in konkreter Form als auch in virtueller Form im Programm betrachtet werden und die Eingabe durch manuelle Auswahl der Befehlskärtchen bzw. der virtuellen Textbausteine bewerkstelligt werden.



Abb. 9: Nachgebaute Umgebung aus Swift Playgrounds

4 Fazit

Der vorgestellte Ablauf zur methodischen und didaktischen Vorgangsweise zur Generierung von Programmiererfahrungen ist der Versuch, stets in enger Verbindung von eigenen Handlungserfahrungen zunehmend und auf sie aufbauend Abstrahierungen zu realisieren. Die vorgestellten Medien (Programmierungsumgebungen bzw. Bodenroboter) wurden zum einen im Sinne des Ermöglichens einer engen Verzahnung der Repräsentationsstufen untereinander und zum anderen im Sinne einer logisch aufeinander aufbauenden Weiterentwicklung informatischer Inhalte (Spiralprinzip) gewählt. Die eingesetzten Medien stellen eine Auswahl von derzeit erhältlichen Werkzeugen dar. Die Entwicklung zukünftiger Medien, deren Einsatz sich noch enger in logischer Abfolge in das Konzept einer aufbauenden Generierung erster Programmiererfahrungen im Sinne des Activity-based-Codings einbetten lässt, ist wünschenswert.

Literatur

- Aebli, H. (2011). *Zwölf Grundformen des Lehrens: eine allgemeine Didaktik auf psychologischer Grundlage*. Klett-Cotta.
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *Journal of Applied Computing and Information Technology* 13(1), 20-29.

- BMBWF (2020). *Volksschul-Lehrplan*. https://www.bmbwf.gv.at/Themen/schule/schulpraxis/lp/lp_vs.html
- Broda, M., & Frank, A. (2015). *Learning Beyond the Screen: Assessing the Impact of Reflective Artificial Intelligence Technology on the Development of Emergent Literacy Skills*. Paper presented at the E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.
- Bruner, J. (1971). *Studien zur Kognitiven Entwicklung*. Klett.
- Bruner, J. (1980). *Der Prozeß der Erziehung* (Sprache und Lernen, Bd. 4). Pädagogischer Verlag.
- Gallenbacher, J. (2017). *Abenteuer Informatik. IT zum Anfassen für alle von 9 bis 99 – vom Navi bis Social Media*. Springer.
- Geier, G. F., & Ebner, M. (2017). Einsatz von OZOBOTs zur informatischen Grundbildung. *Erziehung & Unterricht* 167, 7, 109-113.
- Honegger, B. D. (2017). *Mehr als 0 und 1: Schule in einer digitalisierten Welt*. hep verlag.
- Krauthausen, G., Scherer, P., & Scherer, P. (2017). *Natürliche Differenzierung im Mathematikunterricht: Konzepte und Praxisbeispiele aus der Grundschule*. Kallmeyer in Verbindung mit Klett.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36(2), 1-11.
- Piaget, J., & Szeminska, A. (1975). *Die Entwicklung des Zahlbegriffs beim Kinde*. Klett-Cotta.
- Rabel, M. (2011). Grundvorstellungen in der Informatik. In M. Weigend, M. Thomas, & F. Otte (Hrsg.), *Informatik mit Kopf, Herz und Hand. Praxisbeiträge zur INFOS 2011* (S. 61-70). ZfL-Verlag.
- Sabitzer, B., Antonitsch, P. K., & Pasterk, S. (2014). *Informatics concepts for primary education: preparing children for computational thinking*. Proceedings of the 9th Workshop in Primary and Secondary Computing Education (S. 108-111). ACM Press.
- Savard, A., & Highfield, K. (2015). *Teachers' Talk about Robotics: Where Is the Mathematics?* Proceedings of the Annual Meeting of the Mathematics Education Research Group of Australasia (S. 540-546). MERGA.
- Schwill, A., (2001). Ab wann kann man mit Kindern Informatik machen? Eine Studie über informatische Fähigkeiten von Kindern. In: Keil-Slawik, R. & Magenheimer, J. (Hrsg.), *Informatikunterricht und Medienbildung, INFOS 2001, 9. GI-Fachtagung Informatik und Schule* (S. 13-30). Gesellschaft für Informatik e. V.
- Vom Hofe, R. (1995). *Grundvorstellungen mathematischer Inhalte*. Spektrum Akad. Verlag.
- Wartha, S. (2010). *Aufbau von Grundvorstellungen. Beiträge zum Mathematikunterricht 2010*, 44. Jahrestagung der Gesellschaft für Didaktik der Mathematik vom 08. bis 12. März 2010 in München. Gesellschaft für Didaktik der Mathematik.
- Weigend, M. (2009). Algorithmik in der Grundschule. In B. Koerber (Hrsg.), *Zukunft braucht Herkunft – 25 Jahre »INFOS – Informatik und Schule«*. Gesellschaft für Informatik e.V. <http://dl.gi.de/handle/20.500.12116/20354>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Zorn, I., Trappe, C., Stöckelmayr, K., Kohn, T., & Derndorfer, C. (2013). Interessen und Kompetenzen fördern – Programmieren und kreatives Konstruieren. In *Lehrbuch für Lernen und Lehren mit Technologien* (2. Aufl.). <https://l3t.tugraz.at/index.php/LehrbuchEbner10/article/view/142>