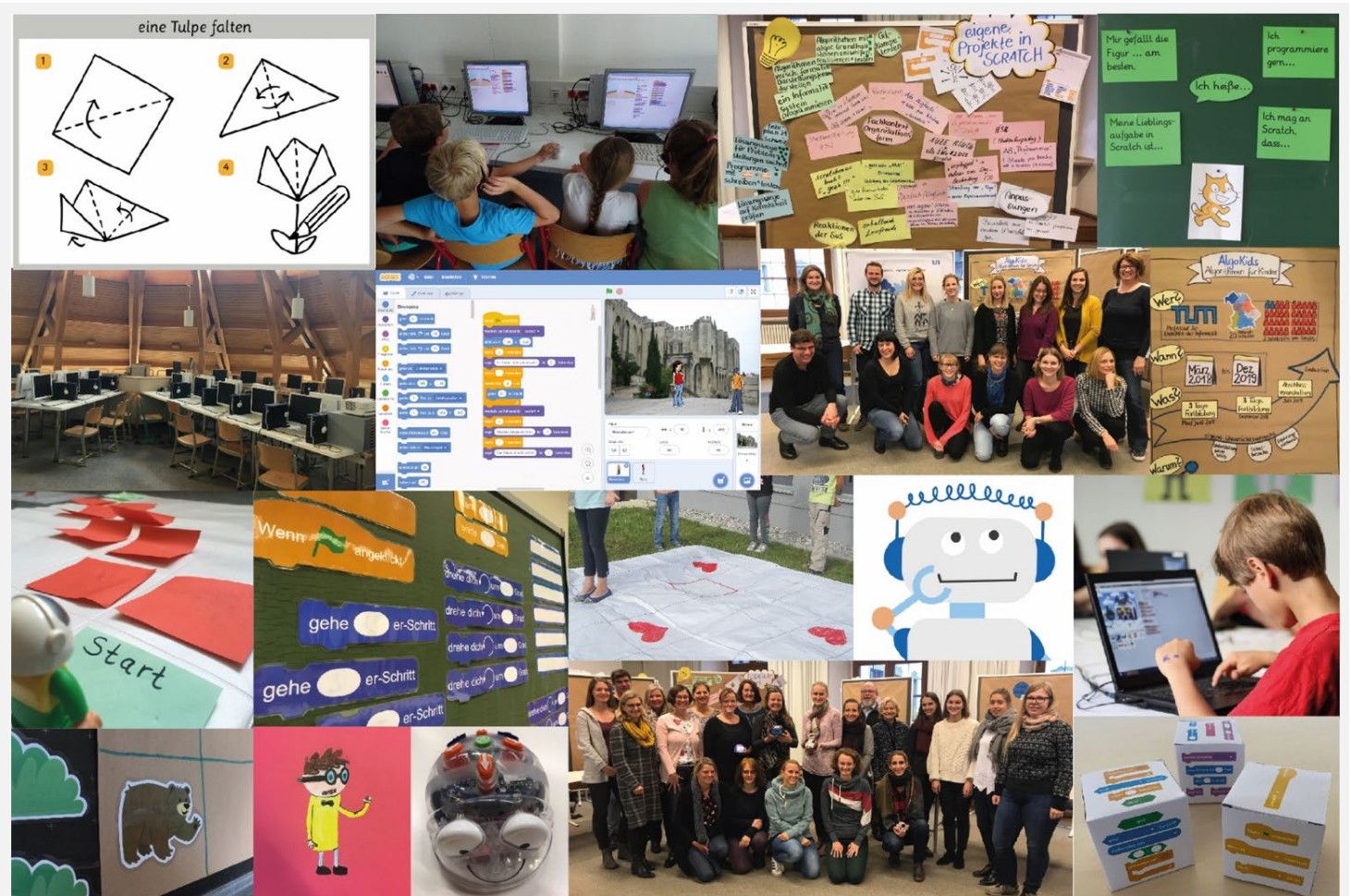


Algorithmen für Kinder (AlgoKids)



**Abschlussbericht eines Kooperationsprojekts
der Technischen Universität München und
des Bayerischen Staatsministeriums für Unterricht und Kultus**

Kurzfassung

Inhalt

1. Einleitung	3
2. Das Projekt im Überblick.....	4
3. Kernelemente der Unterrichtssequenz	6
4. Kernelemente der Lehrerfortbildung.....	8
5. Ausgewählte Ergebnisse der Evaluation	10
6. Zusammenfassung der Ergebnisse.....	15
7. Empfehlungen	16
Publikationen.....	18

1. Einleitung

Sowohl auf nationaler als auch auf internationaler Ebene setzt sich immer mehr die Überzeugung durch, dass die jeweiligen Schulsysteme auf die fortschreitende und umfassende Digitalisierung unserer Lebenswelt mit einer angemessenen Informatikausbildung reagieren müssen, um den Schülerinnen und Schülern auch in Zukunft eine kompetente, selbstbestimmte Teilhabe an dieser technologiezentrierten Welt zu ermöglichen. Zudem kommen immer mehr Regierungen zu der Überzeugung, dass zu einer solchen Ausbildung eine gewisse Grundausbildung im Programmieren gehört.

Der Freistaat Bayern ist in dieser Richtung schon früh aktiv geworden, indem er bereits 2003 ein Pflichtfach Informatik an seinen Gymnasien und kurz darauf (als Informationstechnologie) auch an seinen Realschulen eingeführt hat. Im Rahmen der jüngsten bayerischen Digitalisierungsoffensive wurde dieses Unterrichtsangebot noch weiter ausgebaut sowie ein durchgehendes Fach Informatik auch an den Mittelschulen eingeführt. Damit verfügen alle weiterführenden Schulen in Bayern über verpflichtenden Informatikunterricht.

Weil aber auch für Grundschul Kinder Computer, Tablets und Smartphones mittlerweile zum Alltag gehören, stellt sich die Frage, ob nicht bereits an den Grundschulen mit ersten Schritten im Programmieren begonnen werden muss. Schließlich wachsen die Kinder in einer durch die Digitalisierung geprägten Welt auf und kommen immer früher mit neuen Technologien und Medien in Kontakt. Bekanntermaßen bilden sich geschlechterspezifische Stereotype bzw. Fehlvorstellungen zu technischen Fächern wie der Informatik bereits in der frühen Pubertät aus, wodurch trotz durchaus vorhandener Begabung eine Wahl von Ausbildungsrichtungen oder Berufen in Richtung der MINT-Fächer be- oder sogar verhindert werden könnte. Zahlreiche internationale Initiativen zeigen, dass viele Länder bereits an einer kindgemäßen Informatikausbildung in der Primarstufe arbeiten.

Andererseits kann man Grundschul Kinder nicht beliebig mit zusätzlichen Lerninhalten belasten. Es mehren sich zudem Stimmen, die ein Nachlassen der elementaren Kulturtechniken Lesen, Schreiben und Rechnen beklagen. Man muss also sorgfältig überlegen, ob, in welchem Umfang und mit welcher Zielsetzung man zusätzliche Lerninhalte wie Programmieren in der Grundschule einführen kann und will. Ein vorsichtiger erster Schritt könnte darin bestehen, zunächst das Interesse der Kinder für die Programmierung zu wecken und ihnen ihre eigenen Talente in dieser Richtung bewusst zu machen.

Vor diesem Hintergrund hat sich das Team der Professur für Didaktik der Informatik an der TUM School of Education (im Folgenden kurz TUMDDI) im Jahr 2015 entschlossen, ein Forschungsprojekt zum Programmieren in der Grundschule aufzusetzen. Damit sollte erforscht werden, inwieweit Kinder im Alter von 8-10 Jahren grundlegende algorithmische Konzepte verstehen und einfache Algorithmen implementieren können. Den Mittelpunkt des Projekts bildete eine sehr sorgfältig konzipierte dreitägige Intervention für Grundschul Kinder, die mit ca. 150 Grundschulkindern an der TUM School of Education und am Schülerforschungszentrum Berchtesgadener Land erprobt wurde.

Um die Einsatzmöglichkeiten des Kurses weiter zu erforschen, entstand die Idee für das Projekt „Algorithmen für Kinder“ (im Folgenden kurz „AlgoKids“). Mit diesem Projekt sollte im Detail erforscht werden, in welcher Form eine derartige Intervention zum Programmieren Lernen an den bayerischen Grundschulen dauerhaft implementiert werden könnte und wie man die Lehrkräfte in die Lage versetzen könnte, die Unterrichtssequenz erfolgreich zu unterrichten. Das Projekt wurde mit organisatorischer Unterstützung und finanzieller Förderung des Staatsministeriums als offizieller Schulversuch durchgeführt. Diese Kurzfassung des Abschlussberichts fasst die Planung, den Verlauf und die Auswertung des Projekts zusammen.

2. Das Projekt im Überblick

Als relativ junge Disziplin weist die Informatik im Gegensatz zu den anderen Schulfächern immer noch ein großes Defizit an empirisch fundierter Unterrichtsforschung auf. Dies trifft insbesondere auf die frühkindliche Bildung sowie die Bildung in der Primarstufe zu. Das Projekt *AlgoKids – Algorithmen für Kinder* leistet einen Beitrag dazu, diesem Defizit entgegenzuwirken.

Vorarbeiten an der TUM

Um die Möglichkeiten und Grenzen der Programmierung mit Grundschulkindern auszuloten, entwickelte die TUMDDI den *Programmierzirkus*, einen dreitägigen Programmierkurs speziell für Grundschulkindern der dritten und vierten Klasse. Nach Abschluss der Unterrichtssequenz sollen die Kinder wissen, wie ein Computerprogramm aufgebaut ist, mit einer kindgerechten Programmierumgebung arbeiten und damit kleine multimediale Projekte programmieren können. Die Sequenz wurde mit ca. 150 Grundschulkindern im Alter von 8-10 Jahren außerhalb des Unterrichts erprobt. Erste Ergebnisse zeigen, dass die meisten Kinder dabei überraschend gute Lernerfolge erzielen.

Konzeption

In dem Projekt AlgoKids sollte untersucht werden, wie man die von der TUMDDI entwickelte und pilotierte Unterrichtsintervention im regulären Unterricht der bayerischen Grundschulen implementieren könnte.

Zweitens wollten wir herausfinden, wie man Grundschullehrkräfte ohne informatische Vorbildung so fortbilden kann, dass sie die Themen Algorithmen und Programmierung kompetent unterrichten können. Im Rahmen des Projekts sollte dazu ein Fortbildungskonzept entwickelt und evaluiert werden, auf dessen Grundlage man später alle bayerischen Grundschulen in die Lage versetzen könnte, eine erste Einführung in das Programmieren dauerhaft an ihren Schulen zu etablieren.

Die Projektlaufzeit erstreckte sich auf den Zeitraum von 1. März 2018 bis 31. Dezember (Abb. 2). Ab Februar 2018 konnten sich alle staatlichen Grundschulen auf der Basis eines durch das Ministerium initiierten bayernweiten Ausschreibungsverfahrens bewerben.

Auswahl der Projektschulen

Die Auswahl der Schulen erfolgte durch das Staatsministerium unter Berücksichtigung der für Schulversuche üblichen Kriterien, wie dem Einbezug aller Regierungsbezirke (Abb. 1), der Schulgröße, Erfahrungen im Bereich der digitalen Bildung sowie der digitalen Ausstattung.



Abb. 1 Übersicht über Verteilung der Projektschulen

Die ausgewählten Schulen wurden im April 2018 über ihre Zulassung informiert und aufgefordert, jeweils zwei Lehrkräfte für die erste Fortbildungsveranstaltung anzumelden. Insgesamt 40 Lehrkräfte wurden somit von der TUM fachlich betreut – eine weitere Lehrkraft kam im Laufe des Projekts hinzu. Insgesamt nahmen 39 weibliche und zwei männliche Lehrkräfte am Projekt teil. Das Alter der Lehrkräfte bewegte sich zum Projektstart zwischen *unter dreißig* und *über fünfzig Jahren* (Tabelle 1).

Über die Hälfte der Lehrkräfte (n=27) gab zu Beginn an, überhaupt keine Vorkenntnisse in Informatik zu besitzen, dreizehn hatten Informatik zumindest vorübergehend als Wahl- oder Pflichtfach in der Schule.

Tabelle 1 Altersverteilung der teilnehmenden Lehrkräfte

Alter	Anzahl Lehrkräfte
unter 30 Jahre	8
30-40 Jahre	17
41-50 Jahre	8
über 50 Jahre	8

Fortbildungsmaßnahmen

Für die Fortbildung der beteiligten Lehrkräfte hatte die TUMDDI ein spezielles Konzept entwickelt, das in Zusammenarbeit mit der *Akademie für Lehrerfortbildung und Personalführung (ALP)* realisiert wurde. Um eine bestmögliche Betreuung sicherzustellen, wurden die Lehrkräfte in zwei Gruppen aufgeteilt. Am Anfang standen für jede Gruppe je eine Fortbildungshalbwoche an der ALP im Mai bzw. Juni 2018. Der Fokus lag dabei auf dem Kennenlernen und Ausprobieren des Unterrichtskonzepts der TUM und den darin behandelten informatischen Inhalten.

Die zweite Fortbildungsserie im Umfang je einer Halbwoche für jede Gruppe wurde im Dezember 2018 ebenfalls an der ALP durchgeführt. Hier lag der Schwerpunkt auf einem Erfahrungsaustausch bezüglich der ersten Erprobungen

der Unterrichtssequenz sowie auf der Vertiefung von fachdidaktischen Themen.

Die abschließende dritte Fortbildung fand für beide Gruppen gemeinsam im November 2019 in den Räumlichkeiten der TUM School of Education in München statt. Hier standen fachdidaktische Themen und die Diskussion des Projekts im Vordergrund.

Erprobung an den Projektschulen

Bereits nach der ersten Fortbildung erprobten die Lehrkräfte das Unterrichtskonzept erstmals mit ihren Schülerinnen und Schülern. Dabei waren sie nicht auf ein bestimmtes Umsetzungsformat beschränkt. Für alle in der ursprünglichen Unterrichtssequenz benötigten Materialien wurden von der TUMDDI Druckvorlagen zur Verfügung gestellt. Darüber hinaus erhielten alle Lehrkräfte einen Satz haptische Programmierbefehle sowie eine Filzmatte, auf der sie mit den Befehlen programmieren können.

Unterrichtsbesuche

Während der Erprobung der Unterrichtssequenz sollten alle teilnehmenden Schulen mindestens einmal von der Projektleitung besucht werden. Die Schulbesuche erfolgten nach Abschluss der ersten Fortbildungsserie im Juni 2018 bis zum Ende des Schuljahres 2018/19. Im Rahmen der Besuche wurden insgesamt neunzehn Reflexionsgespräche mit den Lehrkräften geführt. Diese fanden überwiegend gleichzeitig mit beiden Lehrkräften der jeweiligen Projektschule statt

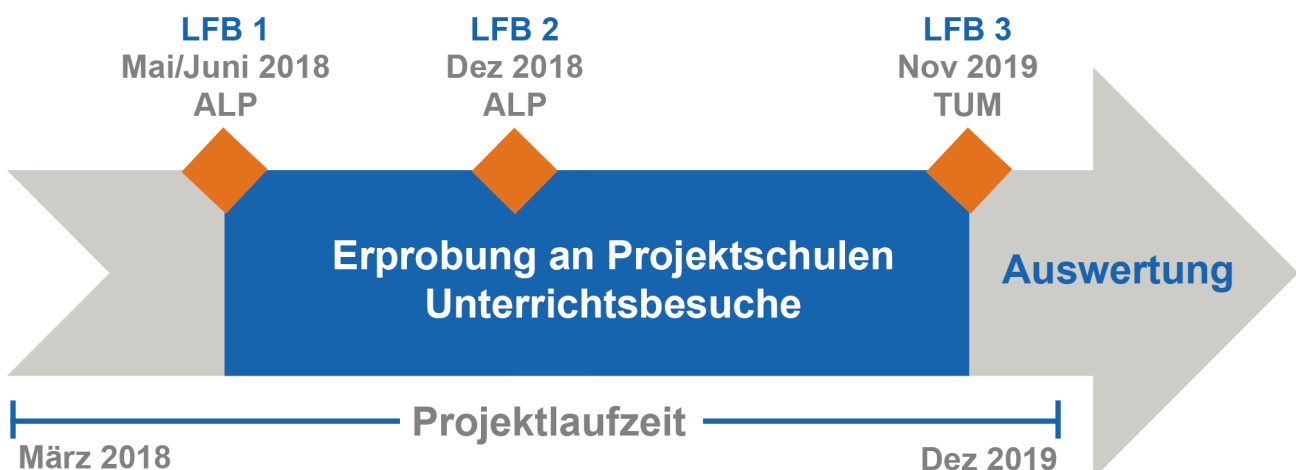


Abb. 2 Übersicht über den Projektverlauf

3. Kernelemente der Unterrichtssequenz

Im Folgenden werden die Kernelemente der Unterrichtssequenz, die von den Lehrkräften im Rahmen des Projekts erprobt wurde, beschrieben. Ausführlichere Beispiele, Anpassungen sowie Erweiterungen können in der Langfassung des Abschlussberichts nachgelesen werden.

Was ist ein Algorithmus?

Da die meisten Grundschüler wenig Vorkenntnisse im Bereich der Programmierung mitbringen, besteht der erste Schritt darin, ihnen eine grundlegende Idee zu vermitteln, wie Algorithmen und Computerprogramme funktionieren. Sie arbeiten *unplugged*, das heißt ohne Computer, und programmieren in Alltagssprache.

Die Schülerinnen und Schüler programmieren zunächst die Lehrkraft mittels mündlicher Anweisungen. Die Lehrkraft spielt einen Roboter, der kleine Aufgaben im Klassenzimmer erfüllen soll, zum Beispiel das Fenster öffnen. Da dieser sich bei zu allgemeinen Anweisungen nicht rührt und nur präzise Befehle befolgt, wird schnell deutlich, dass jeder Schritt in einem Algorithmus verständlich, genau und eindeutig formuliert werden muss. In weiteren Aufgaben wird geübt, Abläufe in natürlicher Sprache zu beschreiben (Abb. 3, Abb. 4).

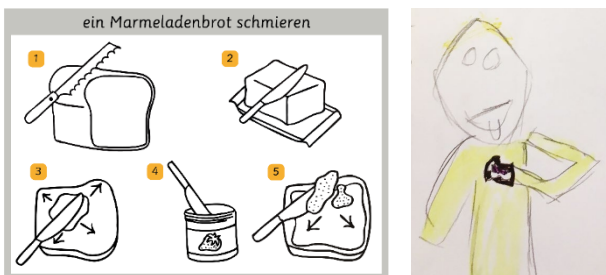


Abb. 3 Bildhafte Anleitung (links) mit Ergänzung (rechts)

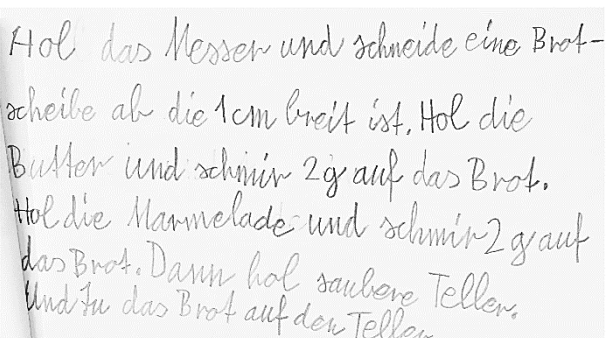


Abb. 4 Anweisungen zum Anrichten eines Marmeladenbrots

Programmieren unplugged

Um das Darstellen von Algorithmen weiter zu vertiefen, programmieren die Schülerinnen und Schüler sich gegenseitig, um verschiedene Aufgaben in einem Parcours zu lösen, zum Beispiel den Weg eines Affen zu seinem Futter (Abb. 5, links). Sobald eine Aufgabe bearbeitet wurde, laufen sie die Lösung im Parcours ab (Abb. 5, rechts). Falls nötig, verbessern sie die Programme gemeinsam.

Neben der Alltagssprache wird hier mit Symbolen und schließlich mit der blockbasierten Programmiersprache *Scratch* gearbeitet (Abb. 6). Um die kognitive Belastung für die Schülerinnen und Schüler möglichst gering zu halten, werden die Befehle der Programmiersprache zunächst *unplugged* verwendet: Für die Bearbeitung der Aufgaben wurden haptische Programmierblöcke angefertigt, die optisch den Programmierbefehlen in *Scratch* entsprechen (Abb. 6, rechts). Diese wurden mit Magneten und Klettverschlüssen ausgestattet, sodass die Schülerinnen und Schüler mit ihnen sowohl an der Tafel als auch auf Filzbahnen programmieren können.



Abb. 5 Wege in einem Parcours beschreiben

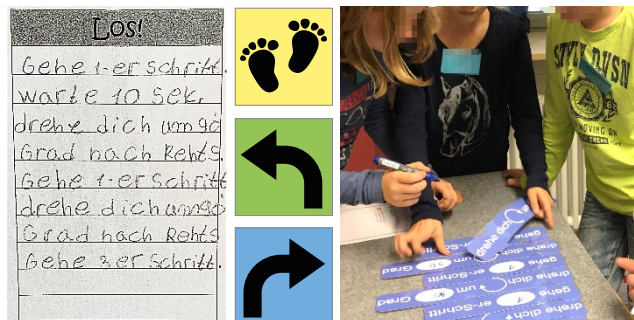


Abb. 6 Verschiedene Wege, um einen Algorithmus darzustellen

Programmieren am Computer

Nach den *Unplugged*-Übungen programmieren die Kinder am Computer in *Scratch*. Damit die Schülerinnen und Schüler sich ganz auf die Bedienung der Programmierumgebung konzentrieren können, bearbeiten sie darin zunächst einige Aufgaben, die bereits *unplugged* gelöst wurden. Im Anschluss bearbeiten die Schülerinnen und Schüler einen Lernzirkel mit zunehmend schwierigen Stationen, in dem die Grundfunktionen von *Scratch* nacheinander thematisiert werden. Ausgehend von Fragen der Bedienung führen die Zirkelkarten (Abb. 7) über einfache Sequenzen hin zu Kontrollstrukturen wie Wiederholungen und bedingte Anweisungen. An jeder Station wird die entsprechende Funktion zunächst schrittweise erklärt und die Schülerinnen und Schüler wiederholen jeden Schritt an ihrem eigenen Rechner. Im Anschluss daran bearbeiten sie eine dazu passende Aufgabe, bei der sie die eingeführte Funktion bzw. Struktur in einem anderen Kontext oder leicht abgewandelt verwenden müssen.

Programme planen

Der letzte Schritt in der Unterrichtssequenz führt die Schülerinnen und Schüler hin zur Planung und Umsetzung eigener Programmideen. In Einzel- oder Partnerarbeit überlegt sich jedes Kind zum Beispiel eine kurze Geschichte, beschreibt diese in einem Drehbuch (Abb. 8) und setzt sie schließlich in *Scratch* um.

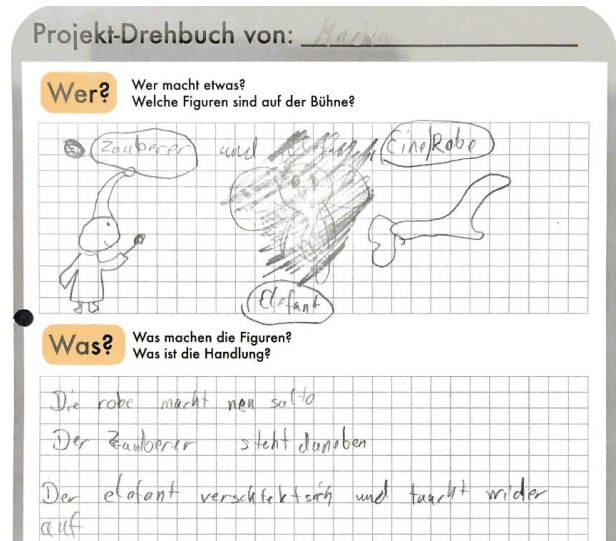


Abb. 8 Ausgefülltes Projekt-Drehbuch

Um die Schülerinnen und Schüler trotz einer offenen Aufgabenstellung nicht zu überfordern, kann man verschiedene Anforderungen vorgeben, die erfüllt werden sollen. Diese können sich einerseits auf den Inhalt der Programme beziehen – zum Beispiel kann ein Gespräch zwischen zwei Figuren oder ein Witz programmiert werden. Andererseits kann die Lehrkraft die zu verwendenden Programmstrukturen vorgeben, zum Beispiel eine Wiederholung oder verschiedene Startereignisse.

Am Ende der Unterrichtssequenz stellen die Schülerinnen und Schüler die eigenen Projekte vor und bekommen Gelegenheit, ihr Vorgehen sowie mögliche Schwierigkeiten zu kommentieren. Hier ist es optimal, wenn sie ihre Ergebnisse der Klasse über den Beamer zeigen können.

2. Begrüßung

Der Zirkusdirektor sagt etwas.

- 1 Öffne in Scratch die Datei „2_Begrüßung“ aus dem Ordner „Zirkel“.
 - 1 Unfass-ab! (Tiger)
 - 2 Begrüßung (Zirkel)
 - 3 Tiger-geräusch (Tiger)
- 2 Klicke auf den Blockbereich „Aussehen“.
- 3 Schiebe den Block insgesamt drei mal nach rechts.
 So soll es aussehen:

- 4 Klicke auf die weißen Felder und ändere den Text des Direktors:
- 5 Klicke auf die grüne Flagge um das Programm zu starten!
- 6 Klicke auf „Datei“ und wähle „Auf deinem Computer speichern“ aus.

Aufgabe 2: Witz

Der Clown soll einen Witz erzählen.

- 1 Öffne in Scratch die Datei „2_Aufgabe_Witz“ aus dem Ordner „Aufgaben“.
 - 1_Aufgabe_Meerschweinchen
 - 2_Aufgabe_Witz-rob
 - 3_Aufgabe_Tigergeräusch
- 2 Lass den Clown einen kurzen Witz erzählen!
- 3 Speichere dein Programm!

Abb. 7 Station des Scratch-Zirkels

4. Kernelemente der Lehrerfortbildung


Da die meisten Lehrkräfte der Grundschule keine oder nur wenige Vorkenntnisse in der Informatik mitbringen, sollten sie im Rahmen von AlgoKids die Chance erhalten, sich mit fachlichen und fachfachdidaktischen Inhalten auseinanderzusetzen. Dafür wurden drei mehrtägige Lehrerfortbildungen durchgeführt, die sich über die Projektlaufzeit verteilten (vgl. Abb. 2). Die theoretischen Grundlagen sowie der genaue Ablauf der einzelnen Fortbildungen kann im entsprechenden Kapitel der Abschlussbericht-Langfassung nachgelesen werden.

Lehrerfortbildung 1

Die erste LFB wurde am Ende des Schuljahres 2017/18 an der ALP durchgeführt. Die Lehrkräfte hatten darin den ersten direkten Kontakt zur TUMDDI. Nach der Fortbildung starteten die ersten Unterrichtsversuche zum Programmieren an den Projektschulen.

Zu Beginn der Fortbildung setzten sich die Lehrkräfte umfangreich mit der Unterrichtssequenz der TUM zum Programmieren in der Grundschule auseinander. Alle Übungen wurden bearbeitet und gemeinsam besprochen. Der zweite Fortbildungstag begann mit einem fachlichen Überblick zum Programmieren in der Schule. Den Lehrkräften wurde aufgezeigt, dass das Programmieren mehr ist als das bloße Erlernen einer Programmiersprache und der Fokus deshalb immer auf den der Programmierung zugrundeliegenden algorithmischen Strukturen liegen sollte: *Anweisung*, *Sequenz*, *Wiederholung* und *bedingte Anweisung*. Nach einer kurzen Erläuterung des Begriffs Algorithmus wurden die einzelnen algorithmischen Grundstrukturen nacheinander behandelt. Jede Struktur wurde zunächst in Pseudocode sowie Blöcken der Programmiersprache Scratch beschrieben (Tabelle 2) und danach in verschiedenen Programmieraufgaben vertieft. Der theoretische Input wurde dabei möglichst kurzgehalten – der Schwerpunkt lag auf dem eigenständigen Arbeiten an den Aufgaben.

Tabelle 2 Anweisung in Alltagssprache, Pseudocode und Scratch

Alltagssprache	Pseudocode	Scratch
Zeichne ein Quadrat	Wiederhole 4 Mal Zeichne eine Linie Drehe Stift um 90° Ende Wiederhole	

Am Ende der ersten Fortbildungshalbwoche wurde thematisiert, wie sich die Lehrkräfte vorstellen können, die Themen Algorithmen und Programmieren in ihrem eigenen Unterricht umzusetzen.

Lehrerfortbildung 2

Die zweite Fortbildung wurde ein halbes Jahr später im Dezember 2019 an der ALP durchgeführt. Im Vorfeld wurde mit den Lehrkräften vereinbart, dass sie bis dahin erste Unterrichtsversuche durchführen sollten.

Zu Beginn der Fortbildung wurde viel Zeit eingeplant, um den Lehrkräften eine umfangreiche Reflexion zu ermöglichen. Um die Programmierkenntnisse der teilnehmenden Lehrkräfte weiter zu festigen, erhielten sie zudem erneut die Gelegenheit, in Scratch zu programmieren und dabei gegebenenfalls auf die Unterstützung der Kursleiter zurückzugreifen. Um gleichzeitig fachdidaktische Themen zu behandeln, standen bei den Aufgaben verschiedene Aufgabenformate in Scratch sowie der Entwicklungsprozess der Lernenden beim Programmieren im Mittelpunkt.

Neben Scratch gibt es noch viele andere Systeme, mit denen Kinder das Programmieren lernen können. Am zweiten Fortbildungstag sollten sich die Lehrkräfte einen Überblick über diese verschaffen und sie selbst ausprobieren, um sich eine eigene Meinung dazu bilden zu können. An verschiedenen Stationen lernten sie weitere Möglichkeiten zum Programmieren im Unterricht kennen: den Einplatinencomputer *Calliope mini*, den *Makey Makey*, den mit Pfeiltasten programmierbaren Bienenroboter

Bluebot, die App *ScratchJr* sowie die programmierbaren Bausätze *LEGO WeDo* und *LEGO EV3*. An jeder Station erhielten die Lehrkräfte grundlegende Informationen zu dem jeweiligen Produkt und bearbeiteten eine einführende Aufgabe (Abb. 9).

LEGO WeDo ca. 150 €

Steckbrief

Altersgruppe:	6-9 Jahre
Preis:	ca. 150€
Stromversorgung:	USB-Kabel
System:	Windows, iOS



Das bietet LEGO WeDo

- Bauen von verschiedenen Figuren mit Lego-Bausteinen
- Bewegung durch Motoren
- Bewegungs- und Kippsensor
- Einfache blockbasierte Programmierung

1 Lassen Sie das Krokodil sein Maul öffnen. Das Krokodil soll danach zubeißen, sobald etwas in seinem Mund ist. Nutzen Sie die folgenden Blöcke.



2 Lassen Sie den Löwen mittels verschiedener Tasteneingaben aufstehen und sich hinlegen.

TIPP: 

Abb. 9 Arbeitsblatt zur Station *LEGO WeDo*

Am letzten Fortbildungstag wurden verschiedene fachdidaktische Theorien vorgestellt und in mehreren Übungen mit den Aufgaben vom Vortag verknüpft. Zum Ende der Fortbildung erarbeiteten die Lehrkräfte Lernstandserhebungen auf der Grundlage verschiedener Kompetenzformulierungen zur Informatik in der Grundschule. Zu einzelnen Formulierungen wurden paarweise Aufgaben inklusive Erwartungshorizont für eine Lernstandserhebung entworfen. Im Anschluss wurden die Aufgaben untereinander ausgetauscht, bearbeitet und korrigiert.

Lehrerfortbildung 3

Die dritte Fortbildung fand im November 2019 in Räumlichkeiten der TU in München statt und ermöglichte allen Lehrkräften einen gemeinsamen Abschluss des Projektes. Am ersten Tag der Fortbildung arbeiteten die Lehrkräfte erneut an Programmieraufgaben in *Scratch*, auf deren Grundlage verschiedene fachdidaktische Themen im Plenum thematisiert werden konnten.

Da im LehrplanPLUS informatische Inhalte nicht explizit verankert sind, bearbeiteten die Lehrkräfte verschiedene Programmieraufgaben, die im Kontext anderer Fächer eingesetzt werden können. Zum Beispiel animierten sie ein kurzes Gedicht in *Scratch* (Abb. 10) oder programmierten geometrische Figuren.

Die drei Spatzen

In einem leeren Haselstrauch
Da sitzen drei Spatzen, Bauch an Bauch.

Der Erich rechts und links der Franz
Und mitten drin der freche Hans.

Sie haben die Augen zu, ganz zu,
Und obendrüber da schneit es, hu!

Sie rücken zusammen dicht an dicht.
So warm wie der Hans hats niemand nicht.

Sie hören alle drei ihrer Herzlein Gepoch.
Und wenn sie nicht weg sind, so sitzen sie noch.

Christian Morgenstern



Erzähler

Franz

Hans

Erich

Abb. 10 Ein Gedicht (links) wird in *Scratch* animiert (rechts)

In Anknüpfung an die zweite Fortbildung, wurde in verschiedenen Aufgaben thematisiert, wie korrekte Programme und positive Programmierpraktiken in *Scratch* aussehen. Beispielsweise wurden den Lehrkräften zwei Lösungen derselben Aufgabe zur Verfügung gestellt, bei denen sie Unterschiede in der Umsetzung finden und Kriterien für die Bewertung ableiten sollten.

Der letzte fachliche Block der Fortbildung setzte den Fokus auf einzelne Aspekte des *Computational Thinking*. In Kleingruppen bearbeiteten die Lehrkräfte fünf verschiedene Stationen und präsentierten diese anschließend im Plenum. Alle Stationen kamen dabei ohne den Einsatz einer Programmiersprache aus.

Zum Abschluss der Fortbildung gaben die Lehrkräfte Feedback und Anregungen zu verschiedenen Themen. Zum Beispiel wurde diskutiert, inwieweit das Programmieren im Lehrplan der Grundschule verankert werden sollte, wie man Lehrkräfte in Zukunft weiterbilden könnte und welche Vernetzungsangebote sie als nützlich erachten. Als Grundlage für die Abschlussdiskussion sammelten die Lehrkräfte über den letzten Tag der Fortbildung auf Plakatwänden Stichpunkte zu den einzelnen Themen. Diese wurden im Plenum aufgegriffen und weiter besprochen.

5. Ausgewählte Ergebnisse der Evaluation¹

Insgesamt sollten durch die Evaluation von Algo-Kids fundierte Erkenntnisse in verschiedenen Domänen generiert werden:

- Ausgangssituation der Lehrkräfte (ED 0)
- Einschätzungen der Lehrkräfte bzgl. der Fortbildungsmaßnahmen (ED 1)
- Kognitive Veränderungen der Lehrkräfte (ED 2)
- Veränderungen im unterrichtspraktischen Handeln der Lehrkräfte (ED 3)
- Veränderungen auf Seiten der Schülerinnen und Schüler (ED 4)

Die Evaluation des Projekts erstreckte sich auf den Zeitraum zwischen den Lehrerfortbildungen zu Beginn und Ende des Projekts. Für die schriftliche Befragung der Lehrkräfte wurden vier Fragebögen entwickelt. Diese Erhebungen richteten sich in Form von Paper-Pencil-Fragebögen an alle teilnehmenden Lehrkräfte². Sie verteilten sich auf insgesamt vier Erhebungszeitpunkte: zu Beginn der ersten Fortbildung, am Ende der ersten Fortbildung, zu Beginn der zweiten Fortbildung und am Ende des Projekts (Abb. 11). Die Ergebnisse der schriftlichen Befragungen wurden durch Rückmeldungen aus den Feedbackgesprächen ergänzt, die im Rahmen der Schulbesuche mit den Lehrkräften geführt wurden. Im Rahmen des Projekts wurden keine Erhebungen mit Schülerinnen und Schülern durchgeführt. Die Lehrkräfte gaben jedoch Einschätzungen zum Lernfortschritt und Verhalten der Lernenden ab.

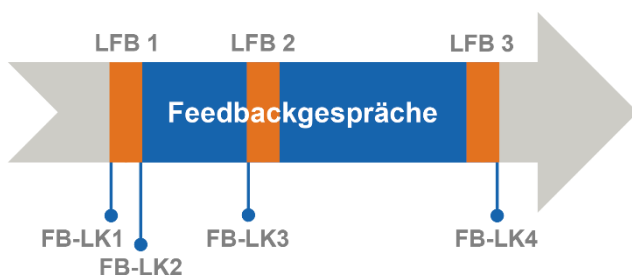


Abb. 11 Anlage der Evaluationsmaßnahmen über das Projekt

¹ Sämtliche Ergebnisse inkl. zusätzlicher Daten, wie z.B. Zitate der Lehrkräfte, können in der Langversion des Abschlussberichts nachgelesen werden.

Erkenntnisdomäne Ausgangssituation (ED 0)

Insgesamt nahmen 39 weibliche und zwei männliche Lehrkräfte am Projekt teil. Das Alter der Lehrkräfte bewegte sich zwischen *unter dreißig Jahren* und *über fünfzig Jahren*. Die Berufserfahrung der Lehrkräfte zu Beginn des Projekts reichte von drei bis 39 Jahren, wobei fast die Hälfte angab, über drei bis zehn Jahre Berufserfahrung zu verfügen (Abb. 12).

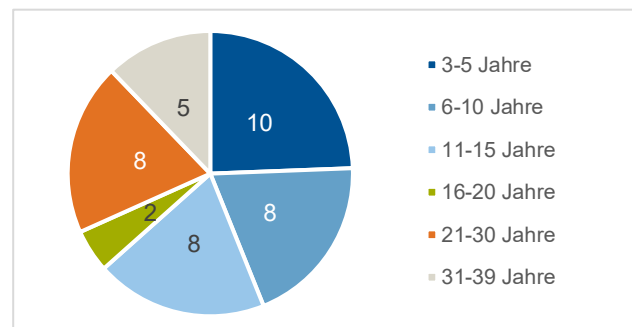


Abb. 12 Berufserfahrung der teilnehmenden Lehrkräfte (n=41)

Die Mehrheit der 41 teilnehmenden Lehrerinnen und Lehrer besaß keine Erfahrung im Programmieren. Bezüglich der IT-Ausstattung der Schulen sowie deren Aktualität äußerten sich die Lehrkräfte zwar relativ ausgeglichen, es zeigte sich jedoch eine Tendenz zu einer stärker negativen Einschätzung.

Erkenntnisdomäne Fortbildungen (ED 1)

Nach der Fortbildung zu Projektbeginn fühlten sich alle Lehrkräfte mindestens teilweise in der Lage, die Inhalte anzuwenden. Bis auf wenige Ausnahmen würden sie sich sogar zutrauen, die Inhalte an Kollegen weiterzugeben. Die meisten Lehrkräfte hielten Dauer und Schwierigkeit der Fortbildung für angemessen. Von den Lehrkräften gaben 33 an, mit der Fortbildung insgesamt *zufrieden* zu sein, fünf *eher zufrieden*. Die Erwartungen hinsichtlich des Nutzens für die Umsetzung an der Schule wurden *teilweise* (n=3), *weitgehend* (n=18) und *voll und ganz* (n=17) erfüllt.

² Die schriftlichen Befragungen wurden vom Bayerischen Staatsministerium für Unterricht und Kultus am 12.11.2018 unter dem Aktenzeichen IV.8-BO7106/113/9 genehmigt.

Am Ende des Projekts wurden die Lehrkräfte zu der Verwendung der Fortbildungsmaterialien befragt (Abb. 13). Es zeigte sich, dass alle befragten Lehrkräfte zumindest einen Teil des Materials im Unterricht verwendeten. Etwa dreiviertel gaben an, es weiterentwickelt zu haben. In den Feedbackgesprächen äußerten sich die Lehrkräfte sehr positiv darüber, dass eine bestehende Unterrichtssequenz komplett zur Verfügung gestellt und in der Fortbildung selbst durchgearbeitet wurde. Die Möglichkeit, sich intensiv mit dem Thema zu beschäftigen und selbst Programmierpraxis zu sammeln, wurde von allen Lehrkräften als unverzichtbar angesehen.

Die Durchführung der Fortbildungen als Präsenzveranstaltungen wurde durchweg positiv bewertet. Besonders bezüglich der *Unplugged*-Aufgaben äußerten viele Zweifel darüber, ob sich ein Selbstlernkurs oder Online-Kurs umsetzen lassen würde. Die Vertiefung von fachdidaktischen Themen in der zweiten Fortbildung sowie die Durchführung der Fortbildungen mit einigen Monaten Abstand, wurde von den Lehrkräften als sinnvoll erachtet

Bei der Diskussion am Projektende kam zudem der Wunsch nach einer längerfristigen Vernetzung über die Projektlaufzeit hinaus auf.

Erkenntnisdomäne Lehrkräfte (ED 2)

Die Evaluation der kognitiven Veränderungen auf Ebene der Lehrkräfte bezieht sich auf deren persönliche Reflexion sowie Selbsteinschätzungen zu verschiedenen Themen. Eine Erhebung des fachlichen oder fachdidaktischen Wissens in Form eines Tests wurde nicht durchgeführt.

Die Lehrerinnen und Lehrer waren motiviert, die informatischen Inhalte im Unterricht umzusetzen und bewerteten die Umsetzung insgesamt als erfolgreich. In der Abschlussbefragung zum Projekt (n=33) gaben 28 Lehrkräfte an, dass sie durch die Umsetzung der Fortbildungsinhalte positive Auswirkungen auf ihre Professionalisierung feststellen konnten. 26 stellten eine positive Veränderung in ihrer Motivation und zwölf in ihrer Arbeitszufriedenheit fest. In Gesprächen merkten einige Lehrkräfte an, dass sie vor der Teilnahme am Projekt überhaupt nicht am Programmieren interessiert waren und es inzwischen als eine persönliche Bereicherung ansehen.

Im Verlauf des Projekts wurde zu verschiedenen Zeitpunkten die Selbstwirksamkeitserwartung der Lehrkräfte hinsichtlich der Umsetzung informatischer Inhalte im Unterricht erhoben. Von 26 Lehrkräften lagen Daten von allen Messpunkten vor (Abb. 14). Hier zeigte sich, dass sie sich durchgängig zutrauten, im Unterricht *unplugged* zu programmieren und die Lernenden für die Thematik zu begeistern. Im Vergleich zum Messzeitpunkt nach der ersten Fortbildung (FB-LK2) zeigte sich, dass die Lehrkräfte sicherer wurden, auf die Fragen ihrer Schülerinnen und Schüler eingehen zu können und mit ihnen am Computer zu programmieren. Interessant ist, dass es einige Lehrkräfte nach ihren eigenen Erprobungen im Unterricht kritischer sehen, selbst den problematischsten Schülerinnen und Schülern die Inhalte vermitteln zu können.

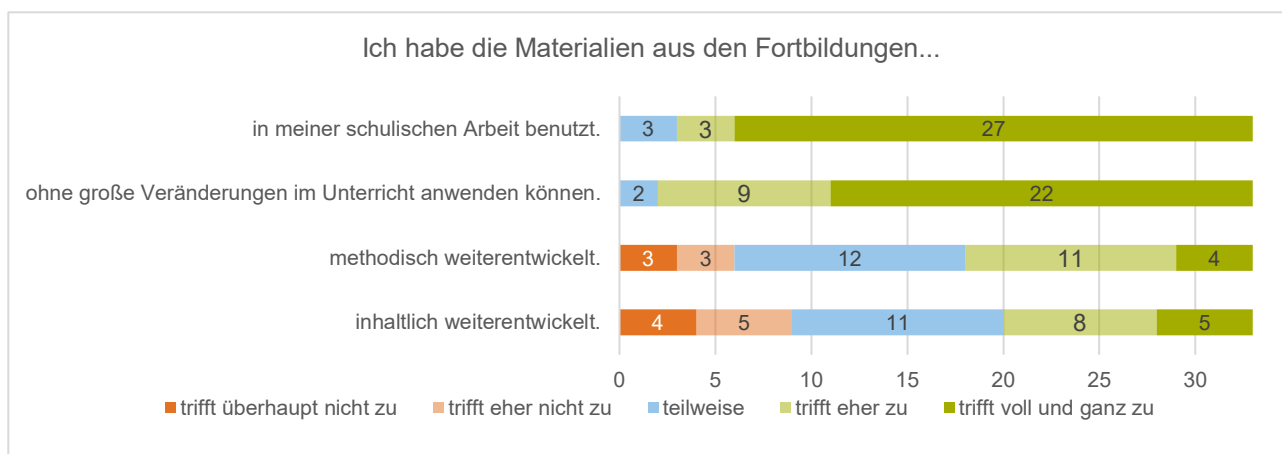


Abb. 13 Angaben zur Verwendung der Fortbildungsmaterialien (n=33)

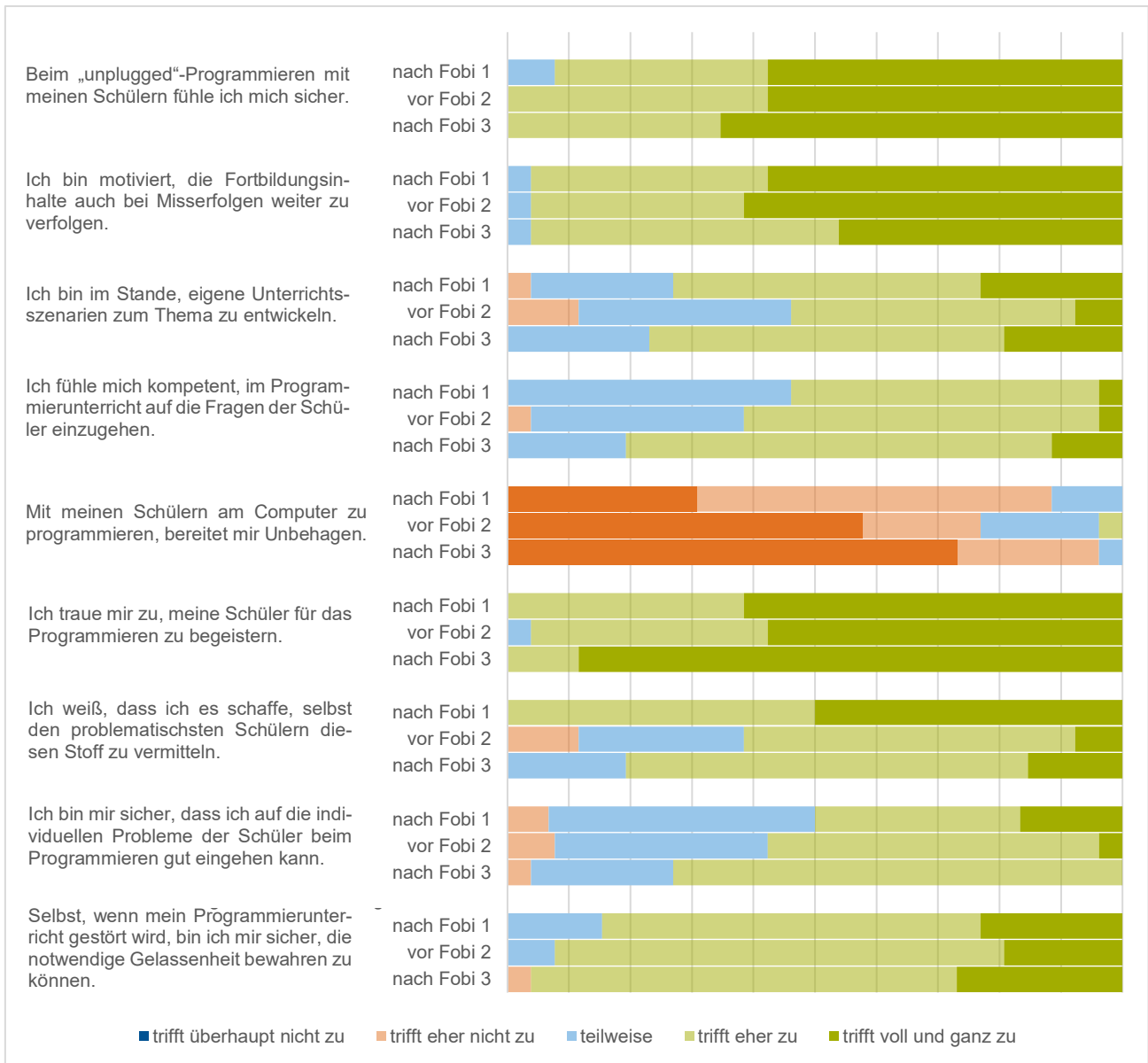


Abb. 14 Angaben zur Umsetzung der informatischen Inhalte im Unterricht (n=26)

An den gleichen drei Messpunkten beurteilten die Lehrkräfte ihre eigenen Fähigkeiten in Bezug auf einzelne Programmierkonstrukte, wie z. B. Wiederholungen oder bedingte Anweisungen, und Programmierpraktiken, wie z.B. das Verbessern von Programmen (Abb. 15). Hier zeigt sich, dass nach den ersten Unterrichtsversuchen im Bereich der Praktiken ein Abfall in der Selbstwirksamkeitserwartung zu verzeichnen war, der sich jedoch bis zum Ende des Projekts wieder aufhob.

In den Angaben, die die Lehrkräfte zum Projektstart und -ende zu ihrem Umgang mit Computern machten, zeigte sich, dass sie sich nach dem Projekt sicherer im Umgang mit Computern

fühlen und sich weniger leicht von auftretenden Problemen aus der Ruhe bringen lassen.

Erkenntnisdomäne Unterricht (ED 3)

Alle teilnehmenden Lehrkräfte erprobten das Programmieren im Unterricht. Auch wenn sich alle Lehrkräfte bei ihrer unterrichtlichen Umsetzung sehr stark an der vorgestellten Sequenz orientierten, zeigten sich Unterschiede in den verschiedenen Implementierungen. Die Unterrichtssequenz wurde an den Projektschulen mit einer Ausnahme mit Schülerinnen und Schülern der Klassenstufe vier erprobt (n=19). Zwei Drittel der Schulen arbeiteten zusätzlich mit Kindern der Klassenstufe drei (n=13).

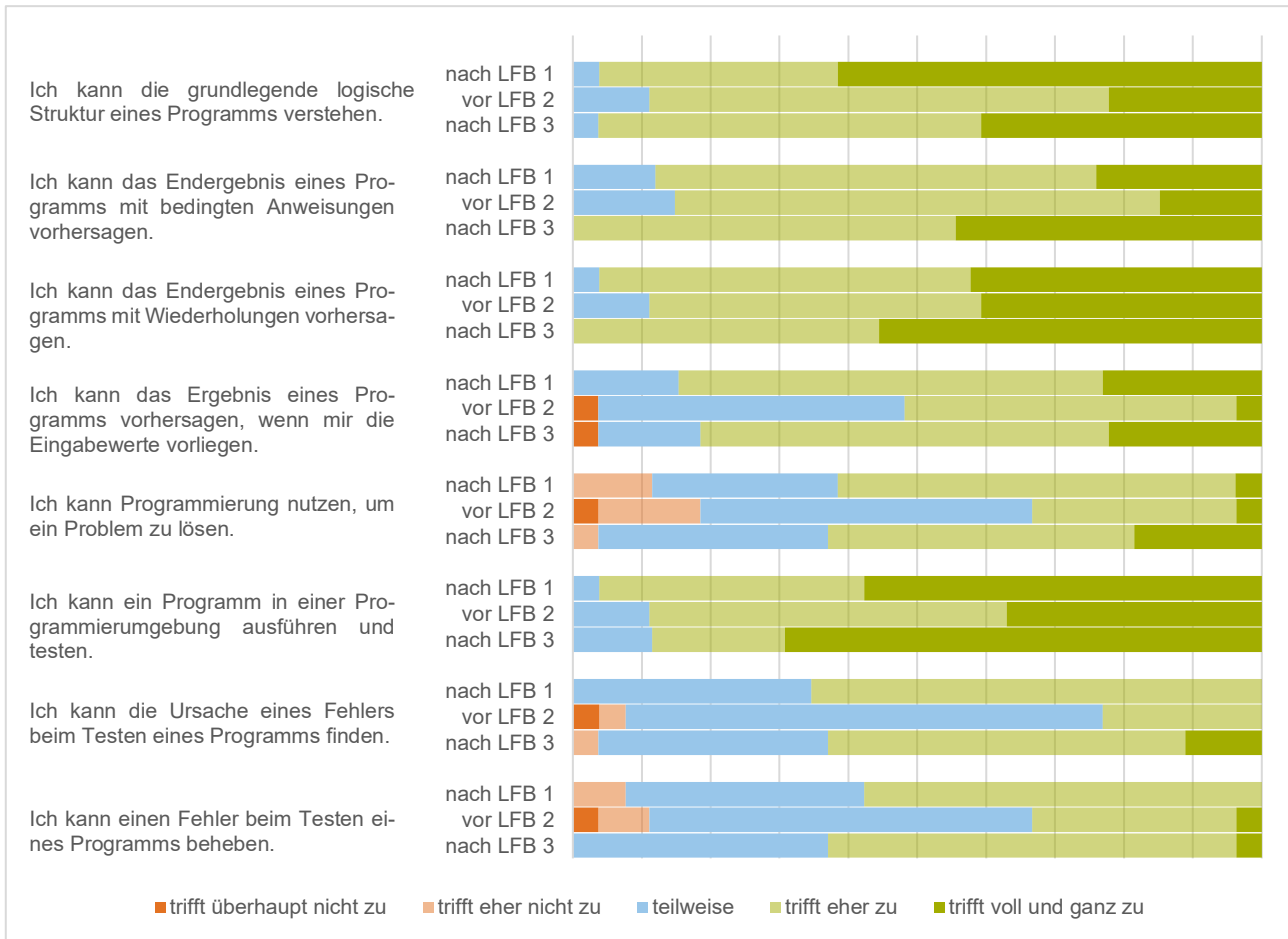


Abb. 15 Angaben zur eigenen Programmierfähigkeit (n=26)

Die Mehrheit der Projektschulen implementierte die Unterrichtssequenz im Klassenverband ($n=17$), ein Viertel der Schulen bot eine AG an ($n=5$), zu der zwischen sechs und sechzehn Kinder zugelassen wurden. An einer Schule wurden Elemente des Konzepts lediglich punktuell mit einzelnen Kindern ausprobiert. Drei Schulen, die im Klassenverbund programmierten, entschieden sich nach der zweiten Fortbildung dazu, *BlueBots* anzuschaffen, und diese in Zukunft zur Vorbereitung der Unterrichtssequenz einzusetzen. Eine Schule arbeitete bereits vor dem Projekt mit den Bienenrobotern. Zwei Schulen, die ausschließlich eine AG anboten, schafften im Laufe des Projekts einen programmierbaren Spielzeugroboter (*Cozmo*) und einen Satz von programmierbaren Bausätzen (*LEGO WeDo*) an. Darüber hinaus arbeitete eine Schule bereits vor Beginn des Projekts im Klassenverbund mit dem *Calliope*.

Die Mehrheit der Lehrkräfte gab an, die Fortbildungsinhalte ohne größere Änderungen angewandt zu haben. Allerdings gab nur die

Hälfte der Befragten an, dass sie die Inhalte in ihre bestehenden schulischen Aktivitäten integrieren konnten. Über die Hälfte der befragten Lehrerinnen und Lehrer gab an, die Inhalte angepasst, weiterentwickelt oder sogar eigene Ideen dazu entwickelt zu haben.

Alle befragten Lehrkräfte gaben an, dass sie die Anwendung der Fortbildungsinhalte für sinnvoll halten. Die Hälfte von ihnen gab an, dass sie durch die Umsetzung der Fortbildungsinhalte positive Auswirkungen auf die Unterrichtsmethodik, Lehrer-Schüler-Interaktionen sowie die Schulentwicklung festgestellt haben. Alle bis auf fünf Lehrkräfte sahen positive Auswirkungen auf die Individualisierung der Lernprozesse (Tabelle 3). Mehr als die Hälfte der Lehrer gab in der Schlussbefragung an, dass die Umsetzung der Fortbildungsinhalte durch fehlende Zeit oder Ausstattung erschwert wurde (Tabelle 4). Für viele Lehrkräfte stellte es eine Schwierigkeit dar, dass informatische Inhalte nicht im Lehrplan verankert sind.

Tabelle 3 Angaben zum Nutzen der Umsetzung (n=33)

Durch die Umsetzung der Fortbildungsinhalte konnte ich positive Auswirkungen feststellen in Bezug auf...	
Unterricht	
Methodik	17
Didaktik	9
(Fach)Inhalte	13
Lehrer-Schüler-Interaktion	17
individualisierte Lernprozesse	28
Schule	
Schulkonzept/Leitbild	10
Schulentwicklung	17
Profil der Schule	9
soziales Klima an der Schule	3

Tabelle 4 Angaben zu Hindernissen in der Umsetzung (n=33)

Die Umsetzung der Fortbildungsinhalte wurde erschwert, da folgende Ressourcen nicht (ausreichend) zur Verfügung standen:	
Zeit	23
Ausstattung	21
Lehr-Lern-Materialien	6
Räumlichkeiten	8
finanzielle Mittel	17

Die Lehrkräfte äußerten sich durchweg positiv über die Heranführung an das Thema, in der die Schülerinnen und Schüler eine erste Idee über das Formulieren von Befehlen und beschreiben von Abläufen erhalten. Neben dem Einstieg über das Programmieren des Lehrers, wurde auch das Beschreiben der bildlichen Anleitungen in natürlicher Sprache als positiv bewertet. Die Hinführung zur Programmierumgebung *Scratch* über die Aufgaben im Parcours wurden von allen Schulen umgesetzt und als essenziell für den Lernprozess der Schülerinnen und Schüler angesehen. Der Parcours wurde dabei auf unterschiedlichste Weise gelegt oder aufgezeichnet, zum Beispiel mit Teppichfliesen, Kreide oder mit Filzstift auf Leintüchern. Einzelne Aufgaben wurden teilweise als zu schwer eingeschätzt, sodass viele Lehrkräfte zusätzlich eigene, leichtere Aufgaben entwickelten. Viele

Lehrkräfte schilderten, dass die Schülerinnen und Schüler sich beim Programmieren ohne Computer (*unplugged*) sehr intensiv mit ihren Programmen auseinandersetzten. Auch wenn die Lehrkräfte diesen Teil der Unterrichtssequenz als wichtig empfanden, waren sie in der Regel nicht der Meinung, dass in Gänze auf die Arbeit am Computer verzichtet werden sollte.

Die bereits erarbeiteten Anleitungen und Übungen im *Scratch*-Lernzirkel waren für die meisten Lehrkräfte eine Erleichterung. Während viele Lehrkräfte den Lernzirkel eins zu eins umsetzten, erweiterten ihn einige Lehrkräfte durch zusätzliche Aufgaben, in denen das bisher erarbeitete nochmals angewandt werden musste.

Erkenntnisdomäne

Schülerinnen und Schüler (ED 4)

Viele der Lehrkräfte stellten durch die Umsetzung der Fortbildungsinhalte eine positive Auswirkung auf ihre Klasse fest (Tabelle 5).

Tabelle 5 Angaben zum Nutzen für die Klasse (n=33)

Durch die Umsetzung der Fortbildungsinhalte konnte ich positive Auswirkungen feststellen in Bezug auf...	
Kompetenzentwicklung der Schülerinnen und Schüler	31
Lernkultur der Klasse	13
Soziales Klima in der Klasse	14

In den Gesprächen zeigte sich ebenfalls, dass die Lehrkräfte das Programmieren für eine sehr nützliche Tätigkeit halten und eine sehr hohe Motivation in ihren Klassen festgestellt hatten – sowohl bei den Jungen als auch den Mädchen. Mehrere Lehrkräfte berichteten in den Feedbackgesprächen jedoch, dass die Jungen im Vergleich zu den Mädchen mehr Vorerfahrungen im Umgang mit Computern aufwiesen und auch eher zuhause programmierten.

Viele Lehrerinnen und Lehrer sahen es als Herausforderung an, den unterschiedlichen Fähigkeiten und Wissensständen der Schülerinnen und Schüler gerecht zu werden.

6. Zusammenfassung der Ergebnisse

Das Projekt *AlgoKids – Algorithmen für Kinder* ging von einer mehrperspektivischen Zielvorgabe aus: Das Unterrichtskonzept der TUM zum Programmieren in der dritten und vierten Jahrgangsstufe sollte an den Projektschulen erprobt und ggf. erweitert werden. Darüber hinaus sollte ein Fortbildungskonzept für Grundschullehrkräfte erarbeitet sowie umgesetzt werden. Während der 22-monatigen Projektlaufzeit wurde zudem untersucht, inwieweit sich die beteiligten Lehrkräfte das Unterrichten der informatischen Inhalte zutrauen und welchen Nutzen sie für ihre Schülerinnen und Schüler sehen. Insgesamt kann man feststellen, dass wir diese Ziele in zufriedenstellendem Maß erreicht haben. Im Folgenden fassen wir die wichtigsten Ergebnisse zu den einzelnen Fragestellungen zusammen.

Unterrichtssequenz

- Die Grundstruktur der ursprünglichen Unterrichtssequenz hat sich an den Projektschulen gut bewährt. Auf Basis der Rückmeldungen und Erweiterungen der Lehrkräfte wurde das Konzept weiter überarbeitet und durch Good-Practice-Beispiele ergänzt.
- Die Lehrerinnen und Lehrer schätzten die kindgerechte Aufbereitung der informatischen Inhalte und die Möglichkeiten für individualisiertes Lernen.
- Die *Unplugged*-Übungen wurden von allen Lehrkräften als notwendig erachtet – eine Umsetzung ausschließlich ohne Programmieren am Computer hielten sie jedoch nicht für sinnvoll.
- Die Lehrkräfte schilderten eine anhaltend hohe Motivation der Kinder beim Programmieren – sowohl der Jungen als auch der Mädchen.
- Das Programmieren in der Grundschule hielten die Lehrkräfte für sehr sinnvoll. Sie berichteten von einem Kompetenzzuwachs ihrer Schülerinnen und Schüler beim Programmieren und

verzeichneten teilweise auch eine Veränderung in deren Arbeitsweise.

- Besonders erfreulich fanden die Lehrkräfte, wie gut die Schülerinnen und Schüler beim Programmieren zusammenarbeiteten und sich gegenseitig unterstützten.

Das Fortbildungskonzept

- Die Lehrkräfte fanden es gut, dass unser Fortbildungskonzept sowohl Erkenntnisse aus der Informatikdidaktik als auch aus der Erwachsenenbildung berücksichtigte. Dies bewährte sich in der Umsetzung insgesamt sehr gut.
- Die Lehrerinnen und Lehrer betonten die Notwendigkeit einer ausreichenden Fortbildung, um die erforderlichen Kenntnisse für die Umsetzung der Inhalte im Unterricht zu erwerben.
- Besonders positiv wurde angesehen, dass im Rahmen der Fortbildung eigene Programmierpraxis und zwischen den einzelnen Fortbildungen erste Erfahrungen in der Umsetzung an der Schule gesammelt werden konnten.
- Die teilnehmenden Lehrkräfte konnten sich nicht vorstellen, dass die Fortbildungen ausschließlich online durchgeführt werden könnten.
- Die Ergebnisse der Evaluation zeigen, dass die Lehrkräfte sich das Unterrichten der Inhalte durch entsprechende Fortbildung zutrauen.

Kritische Aspekte

- Als problematisch für die Umsetzung wurde die hohe Schülerzahl einiger Klassen geschildert.
- Die fehlende Verankerung der Inhalte und Kompetenzen im Lehrplan bereitete den Lehrerinnen und Lehrern Sorgen.
- Die technische Ausstattung der teilnehmenden Schulen war sehr unterschiedlich. In vielen Fällen waren die Geräte nur bedingt einsatztauglich.
- Oft fehlte eine ausreichende Systembetreuung.

7. Empfehlungen

Auf der Grundlage der Ergebnisse des Projekts können wir dem Staatsministerium guten Gewissens empfehlen, Programmieren in der Grundschule nach diesem Konzept an allen bayerischen Grundschulen einzuführen.

Kurz- und mittelfristige Maßnahmen

Aus den Erfahrungen, die im Rahmen von AlgoKids gemacht wurden, lassen sich folgende Empfehlungen für kurz- und mittelfristige Maßnahmen ableiten.

Handreichung für Lehrerinnen und Lehrer

Wir schlagen vor, das *Staatsinstitut für Schulqualität und Bildungsforschung* (ISB) mit der Erstellung einer Handreichung zum Programmieren in der Grundschule zu beauftragen. Die Handreichung sollte die Materialien und Erfahrungen aus dem Projekt AlgoKids berücksichtigen, die sich als erfolgreich erwiesen haben (Tabelle 6).

Lehrerfortbildung

Gleichzeitig mit der Erstellung der Handreichung sollte die ALP mit der Entwicklung und Umsetzung eines Fortbildungskonzepts zum Programmieren in der Grundschule beauftragt werden. Die Maßnahme sollte in Präsenz stattfinden und mindestens zwei Fortbildungshalbwochen je Lehrkraft umfassen. Zwischen den beiden Halbwochen sollte ein ausreichender Zeitraum für eigene Unterrichtserfahrungen im Programmieren liegen. Für die beiden Veranstaltungen empfehlen wir die Themen und Methoden aus dem Fortbildungskonzept von AlgoKids (Tabelle 7).

An jeder Schule sollten mindestens zwei Lehrkräfte nach diesem Konzept fortgebildet werden. Sie sind auch Ansprechpartner für andere Lehrkräfte an der jeweiligen Schule. Dazu müsste vermutlich auf ein Multiplikatorensystem zurückgegriffen werden. Die Ausbildung der Multiplikatoren müsste allerdings im Vergleich zu den anderen Teilnehmern wesentlich umfangreicher und intensiver ausfallen.

IT-Ausstattung und Systembetreuung

Damit die Einführung informatischer Inhalte in der Grundschule gelingen kann, sollten angemessene technische Rahmenbedingungen geschaffen werden. Zum Programmieren sollte jede Grundschule mindestens über einen Klassensatz (d.h. ca. 25) Laptops oder Tablets mit Internetzugang verfügen. Die Geräte müssen regelmäßig ausreichend gewartet werden, wozu Lehrkräfte grundsätzlich nicht herangezogen werden sollten, weil ihnen sowohl die nötige Zeit als auch die notwendigen Kenntnisse fehlen.

Langfristige Maßnahmen

Neben diesen kurz- und mittelfristigen Maßnahmen sind zusätzliche langfristige Aktivitäten notwendig, um die Informatik an den Grundschulen nachhaltig abzusichern.

Lehrplanentwicklung

Informatik als zusätzliches Schulfach in der Grundschule einzuführen, ist in Anbetracht des Mangels an qualifizierten Lehrkräften wohl nur bedingt möglich und sinnvoll. Informatische Inhalte sollten dennoch bei der nächsten Revision des LehrplanPLUS explizit im Lehrplan verankert werden, beispielsweise in den Fächern Deutsch, Mathematik und dem Sachunterricht. Sobald informatische Inhalte im Lehrplan verankert sind, sollten die Schulbuchverlage zur Erarbeitung von Unterrichtswerken zur Informatik in der Grundschule aufgefordert werden.

Lehrerausbildung

Um informatische Bildung in der Grundschule langfristig zu stabilisieren, muss Informatik als vollwertiges Fach in die erste Phase der Lehrerausbildung integriert werden. Zu diesem Zweck sollte zunächst ein Didaktikfach Informatik für das Lehramt Grundschule in der LPO I verankert werden. In einem zweiten Schritt sollte auch ein Unterrichtsfach Informatik in die Liste der Fächerverbindungen aufgenommen werden, die für Lehramtsstudierende zulässig sind.

Tabelle 6 Elemente des Unterrichtskonzeptes

Thema	Kurzbeschreibung	Umfang
Grundlagen von Algorithmen und Programmen	<ul style="list-style-type: none"> Beschreiben von Algorithmen und ihren Abläufen in Alltagssprache „Programmieren“ von natürlichen Personen (Lehrkraft, Schülerinnen und Schüler) 	ca. 1 Doppelstunde
Programmieren unplugged	<ul style="list-style-type: none"> Beschreiben von Abläufen mit Symbolen, z.B. von Wegen in einem Raster Beschreiben von Abläufen mit haptischen Darstellungen von Programmieranweisungen Anwenden und Zusammensetzen von algorithmischen Strukturelementen 	ca. 1-2 Doppelstunden
Programmieren am Computer	<ul style="list-style-type: none"> Programmieren eines „echten“ Informatiksystems mit einer blockbasierten Programmiersprache, z.B. <i>Scratch</i>, <i>Blockly</i> oder <i>Snap</i>. Anwenden von Grundfunktionen des Systems und Implementierung von algorithmischen Strukturelementen 	ca. 2-3 Doppelstunden
Aufgaben analysieren und Programme planen	<ul style="list-style-type: none"> Analyse von schwierigeren Aufgabenstellungen Umfangreichere Abläufe und Algorithmen entwerfen Implementieren von umfangreicheren Algorithmen mit einer blockbasierten Programmiersprache 	ca. 1-2 Doppelstunden

Tabelle 7 Struktur des Fortbildungskonzeptes

	Themenschwerpunkte
Fortbildungshalb-woche 1	<ul style="list-style-type: none"> Ausprobieren einer Unterrichtssequenz in der „Schüler(innen)rolle“ Diskussion der fachlichen Grundlagen, insbesondere der Fachkonzepte Praktisches Programmieren: Anwendung der algorithmischen Strukturelemente Kennenlernen verschiedener Aufgabentypen beim Programmieren (z.B. Programme analysieren, Programme ändern, Programme erstellen)
Fortbildungshalb-woche 2	<ul style="list-style-type: none"> Erfahrungsaustausch über die eigenen Erfahrungen Praktisches Programmieren: Unterstützung von Lernprozessen der Schülerinnen und Schüler, Finden und Verbessern von Fehlern Einblick in weitere Möglichkeiten zum Programmieren in der Grundschule
zusätzliche Angebote	<ul style="list-style-type: none"> Programmieren mit anderen Systemen (aktuell z.B. <i>Calliope mini</i>, <i>Makey Makey</i>, <i>Ozobot</i>³, <i>Dash & Dot</i>⁴)

³ <https://ozobot-deutschland.de/>

⁴ <https://www.makewonder.de/dash/>

Publikationen

Die TUMDDI hat folgende Publikationen zum Thema „Programmieren in der Grundschule“ veröffentlicht:

Geldreich, K. & Hubwieser, P. (2020). Programming in Primary Schools: Teaching on the Edge of formal and non-formal Learning. In: Giannakos M. (eds) Non-Formal and Informal Science Learning in the ICT Era. Lecture Notes in Educational Technology. Springer, Singapore.

Geldreich, K. & Hubwieser, P. (2020). Implementierung einer Unterrichtssequenz zu Algorithmen und Programmierung in der Grundschule. Eine qualitative Interviewstudie mit Grundschullehrkräften. In: Thumel, M., Kammerl, R. & Irion, T. (Eds.), Digitale Bildung im Grundschulalter. Grundsatzfragen zum Primat des Pädagogischen, kopaed.

Geldreich, K. (2019). Programmieren in der Grundschule – aber wie? In: SCHULWELT NRW – Zeitschrift für Lehrerinnen, Lehrer und Schulleitungen in NRW, 1. Jg, 12/2019. S. 6-8. ISSN 2626-823X.

Geldreich, K., Simon, A. & Starke, E. (2019). Which Perceptions Do Primary School Children Have about Programming? In: Proceedings of the 14th Workshop on Primary and Secondary Computing Education (WiPSCE '19), pp. 1-7.

Simon, A., Geldreich, K. & Hubwieser, P. (2019). How to Transform Programming Processes to Graphical Visualizations. In: Proceedings of the 14th Workshop on Primary and Secondary Computing Education (WiPSCE '19), pp. 48-56. DOI: doi.org/10.1145/3361721.3361723.

Geldreich, K., Talbot, M. & Hubwieser, P. (2019). Aufgabe ist nicht gleich Aufgabe – vielfältige Aufgabentypen bewusst in Scratch einsetzen. In: Informatik für alle (INFOS 2019). Bonn: Köllen, pp. 181-190.

Geldreich, K., Simon, A. & Hubwieser, P. (2019). A Design-Based Research Approach for introducing Algorithmics and Programming to Bavarian Primary Schools: Theoretical Foundation and Didactic Implementation. MedienPädagogik: Zeitschrift für Theorie Und Praxis Der Medienbildung 33 (Februar), 53-75.

Geldreich, K., Talbot, M. & Hubwieser, P. (2018). Off to new shores: Preparing Primary School Teachers for Teaching Algorithmics and Programming. In Proceedings of the 13th Workshop on Primary and Secondary Computing Education (WiPSCE '18), pp. 139-144.

Funke, A. & Geldreich, K. (2017). Measurement and Visualization of Programming Processes of Primary School Students in Scratch. In Proceedings of the 12th Workshop on Primary and Secondary Computing Education (WiPSCE '17), Erik Barendsen and Peter Hubwieser (Eds.). ACM, pp. 101-102. DOI: doi.org/10.1145/3137065.3137086.

Funke, A. & Geldreich, K. (2017). Gender Differences in Scratch Programs of Primary School Children. In Proceedings of the 12th Workshop on Primary and Secondary Computing Education (WiPSCE '17), Erik Barendsen and Peter Hubwieser (Eds.). ACM, pp. 57-64. DOI: doi.org/10.1145/3137065.3137067.

Geldreich, K., Funke, A. & Hubwieser, P. (2017). Willkommen im Programmierzirkus - Ein Programmierkurs für Grundschulen. In: Informatische Bildung zum Verstehen und Gestalten der digitalen Welt (INFOS 2017). Bonn: Köllen, pp. 327-334.

Funke, A., Geldreich, K. & Hubwieser, P. (2017). Analysis of Scratch Projects of an Introductory Programming Course for Primary School Students. In Proceedings of the IEEE Global Engineering Education Conference (EDUCON '17), IEEE Press, pp. 1233-1240.

Funke, A., Geldreich, K. & Hubwieser, P. (2016). Primary School Teachers' Opinions About Early Computer Science Education. In Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16), ACM, pp. 135-139.

Geldreich, K., Funke, A. & Hubwieser, P. (2016). A Programming Circus for Primary Schools. In Proceedings of the 9th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (ISSEP 2016), pp. 46-47.

Abbildungen

Die Abbildungen im vorliegenden Bericht sind im Rahmen des Projekts „Algorithmen für Kinder“ bzw. den Vorarbeiten der TUM entstanden. Die Verwendung einer der Abbildungen außerhalb des Abschlussberichts muss mit der TUMDDI abgesprochen werden.

Verschiedene Abbildungen enthalten Screen-shots der Oberfläche von *Scratch* (CC-BY-SA-4.0 Creative Commons), eines Projekts der *Scratch Foundation* und der *Lifelong Kindergarten Group* am *MIT Media Lab*.