

Variablen

MASTERARBEIT

Informatikunterricht im Zyklus 2: Variablen

Eine Design-Based-Research Studie

Eingereicht von:
Lea Gisler

Betreut durch:
Prof. Dr. Eva Marinus

31. Juli 2024

DOI: 10.5281/zenodo.14185489

Abstract

In dieser Masterarbeit wurde der Informatikunterricht in der Primarstufe in Bezug auf das Programmierkonzept *Variable*. Dieses bedeutende Konzept ist insbesondere für junge Lernende schwierig zu Verstehen. Um das Konzept zu veranschaulichen, können Analogien verwendet werden. Durch diese Vereinfachung können aber auch falsche Vorstellungen erzeugt werden. Ziel dieser Arbeit war, herauszufinden, welche Analogien im Informatikunterricht verwendet werden können, um Variablen Lernenden im Alter von 10–12 Jahren zu erklären. Dafür wurden innerhalb eines Design-Based-Research-(DBR-)Prozesses drei Untersuchungen durchgeführt. Zuerst wurden Lehrmittel und Forschungsbeiträge sowie die Meinung von Personen mit Expertenwissen in Bezug auf die Verwendung von Analogien für Variablen untersucht. Die beiden Untersuchungen zeigten Hinweise darauf, dass eine Alltagsanalogie beim Erklären des Programmierkonzepts *Variable* hilfreich sein könnte. Folglich wurden die Beschreibungen über die Alltagsvariablen Zeit, Temperatur und Geschwindigkeit in einem Interview mit Lernenden gesammelt und mit den Eigenschaften von Variablen verglichen. Die Ergebnisse zeigen eine Überschneidung in der Beschreibung der Temperatur in Bezug auf die Veränderlichkeit von Werten einer Variable. Das sind erste Hinweise, die bedeuten, dass die Analogie «Eine Variable ist wie die Temperatur, weil sich ihre Werte verändern können» hilfreich sein könnte für Lernende im Alter von 10–12 Jahren. Die Arbeit trägt durch die Kombination von theoretischem Hintergrund und empirischen Ergebnissen zur Entwicklung möglicher Design-Prinzipien für den Informatikunterricht über Variablen bei. Allerdings ist der Designprozess nicht abgeschlossen, weshalb weitere Untersuchungen notwendig sind, um die gefundenen Prinzipien zu bestätigen.

Schlüsselwörter: Informatikunterricht, Zyklus 2, Variablen, Analogien, Design-Based-Research

Glossar

DBR	Design-Based Research
CT	Computational Thinking
FV	Fehlvorstellung
FF	Forschungsfrage
U	Untersuchung
DP	Design-Prinzipien

Danksagung

An dieser Stelle möchte ich mich herzlich bei allen Personen bedanken, die zum Gelingen dieser Masterarbeit beigetragen haben.

Mein besonderer Dank gilt meiner Betreuerin Eva Marinus für ihre Unterstützung, wertvollen Ratschläge und die kontinuierliche Motivation während des gesamten Forschungsprozesses. Ihre fachliche Expertise und ihr konstruktives Feedback waren für diese Arbeit von unschätzbarem Wert.

Ein grosser Dank gebührt auch den drei Expertinnen, die sich die Zeit genommen haben, an den Interviews teilzunehmen. Ihre wertvollen Einblicke und ihr Fachwissen haben massgeblich zur Qualität dieser Arbeit beigetragen. Ebenso möchte ich den Lehrpersonen danken, welche mit ihrer Klasse an den Interviews teilgenommen haben.

Nicht zuletzt möchte ich meiner Familie und meinen Freunden danken. Ihre unermüdliche Unterstützung, Geduld und Ermutigung haben mir die nötige Kraft gegeben, diese Herausforderung zu meistern.

Inhaltsverzeichnis

1. Einleitung.....	1
2. Theorie und Forschungsstand	3
2.1 Fachwissenschaftliche Grundlagen der Informatik	3
2.1.1 Programmieren	3
2.1.2 Programmiersprache und Programmierumgebung	3
2.1.3 Programmierkonzepte	4
2.1.4 Programmierkonzept «Variablen»	5
2.1.5 Abgrenzung zu anderen Fachbereichen	8
2.2 Fachdidaktische Grundlagen des Programmierens	8
2.2.1 Computational Thinking	8
2.2.2 Vermittlung von Programmierkompetenzen	8
2.2.3 Didaktische Rekonstruktion	9
2.2.4 Vorstellungen von Lernenden	11
2.2.5 Analogie in der Didaktik	11
2.2.6 Analogien in der Informatik	14
2.2.7 Zusammenfassung	14
2.3 Wissenschaftliche Erkenntnisse zur Vermittlung von Variablen	15
2.3.1 Fehlvorstellungen zu Variablen	15
2.3.2 Mathematik und Programmierkonzept «Variablen»	17
2.3.3 Analogien für Variablen	17
2.3.4 Rollen von Variablen	19
2.3.5 Kompetenzaufbau für Variablen	21
2.3.6 Diverse Befunde	22
2.3.7 Zusammenfassung	23
3. Design und Methodik der Untersuchungen	24

3.1	Forschungslücke	24
3.2	Design-Based-Research	24
3.3	Planung/Ablauf der Untersuchung.....	26
3.3.1	Focus	26
3.3.2	Understand	26
3.3.3	Define	27
3.3.4	Conceive	27
3.4	Didaktische Rekonstruktion «Variablen».....	27
3.5	Verwendung von KI-Tools	28
4.	Untersuchung 1: Analyse Lehrmittel und Forschungsbeiträge	30
4.1	Forschungsfrage und Erwartungen	30
4.2	Stichprobe.....	31
4.3	Untersuchungsmaterialien.....	33
4.3.1	Analogien für Variablen	33
4.3.2	Rollen von Variablen	33
4.3.3	Aktivitäten	34
4.4	Prozedur	34
4.5	Ablauf der Datenauswertung.....	34
4.6	Ergebnisse.....	34
4.6.1	Verwendete Analogien für Variablen	35
4.6.2	Vorkommende Rollen von Variablen	37
4.6.3	Aktivitäten für Variablen	38
4.7	Zwischendiskussion und Fazit.....	41
4.7.1	Beantwortung von FF1	42
4.7.2	Implikationen.....	43
5.	Untersuchung 2: Experten- und Expertinneninterviews.....	44
5.1	Forschungsfrage und Erwartungen	44

5.2 Stichprobe.....	45
5.3 Untersuchungsmaterialien.....	45
5.4 Prozedur	46
5.5 Ablauf der Datenauswertung.....	46
5.6 Ergebnisse	47
5.6.1 Analogien für Variablen	47
5.6.2 Aktivitäten für Variablen	49
5.6.3 Didaktische Hinweise für Variablen	51
5.7 Zwischendiskussion und Fazit.....	52
5.7.1 Beantwortung von FF2	53
5.7.2 Implikationen	54
6. Untersuchung 3: Kinderinterviews.....	56
6.1 Forschungsstand zur Vorstellung von Alltagsvariablen	56
6.1.1 Temperatur	57
6.1.2 Zeit.....	57
6.1.3 Geschwindigkeit	58
6.2 Forschungsfragen	59
6.3 Stichprobe.....	59
6.4 Untersuchungsmaterial	61
6.4.1 Aufbau des Interviewleitfadens.....	61
6.4.2 Scratchprogramme.....	62
6.4.3 Probeinterview	64
6.5 Prozedur	66
6.6 Ablauf der Datenauswertung.....	66
6.7 Ergebnisse.....	67
6.7.1 Beschreibung der Lernenden über Zeit	68
6.7.2 Beschreibung der Lernenden über Temperatur	69

6.7.3	Beschreibung der Lernenden über Geschwindigkeit	71
6.7.4	Beschreibung der Lernenden über Variablen	72
6.7.5	Zusätzliche Befunde.....	74
6.8	Zwischendiskussion und Fazit.....	75
6.8.1	Beantwortung von FF3	75
6.8.2	Beantwortung von FF4	77
6.8.3	Implikationen	78
7.	Diskussion	80
8.	Reflexion und Ausblick.....	84
8.1	Implikationen für die Fachdidaktik Informatik.....	84
8.2	Diskussion des methodischen Vorgehens	84
8.2.1	Analyse Lehrmittel und Forschungsbeiträge.....	84
8.2.2	Experten- und Expertinneninterviews	85
8.2.3	Kinderinterviews	86
8.3	Zukünftige Forschung und Empfehlungen	86
8.4	Fazit.....	87
9.	Literaturverzeichnis.....	88
10.	Anhang.....	99

1. Einleitung

Seit der Einführung des Lehrplans 21 ist das Fach «Medien & Informatik» in der Schweiz auf der Lektionentafel enthalten. Im Modullehrplan Medien und Informatik beschreibt die Kompetenz MI2.2, dass die Lernenden einfache Problemstellungen analysieren, mögliche Lösungswege beschreiben und in Programmen umsetzen können sollen (D-EDK, 2016a). In der Praxis bedeutet dies, dass sich Lernenden eine Programmiersprache mithilfe einer Programmierlernumgebung, wie Scratch oder xLogo, aneignen. Durch das Erlernen einer solchen Programmiersprache werden Kompetenzen wie das Planen von Lösungsstrategien, das abstraktes Denken oder auch das Herangehen an komplexe Probleme gefördert (Papadakis et al., 2016; Wang et al., 2016), welche im 21. Jahrhundert zu relevanten Kompetenzen zählen. (Bers, 2019). Aus diesen Gründen ist das Interesse an der Vermittlung von Programmierkenntnissen im Bildungsbereich, insbesondere bei jungen Lernenden, gross (Guggemos, 2021; Macrides et al., 2022). Ein Fokus in der Informatikdidaktik liegt auf der Identifikation von Schwierigkeiten im Lernprozess des Programmierens (Qian & Lehman, 2018). Studien weisen darauf hin, dass unter anderem das Verständnis des Programmierkonzepts Variablen Lernende vor Herausforderungen stellt (Grover et al., 2019; Hermans et al., 2018; Kohn, 2017). Das zeigt sich auch durch zahlreiche Bezüge zu Variablen in möglichen Fehlvorstellungen von Lernenden (Sorva, 2018). Eine verbreitete Fehlvorstellung ist beispielsweise, dass eine Variable mehrere Werte gleichzeitig speichern und sich an alte Werte «erinnern» kann (Sorva, 2013). Solche falschen Vorstellungen können zu einem fehlerhaften Code und nicht lauffähigen Programmen führen, was für Lernende frustrierend sein kann (Sorva, 2018). Darüber hinaus zählt ein grundlegendes Verständnis über Variablen zu den Schlüsselkompetenzen für das Lesen und Verstehen von Code (Corney et al., 2011; Lister et al., 2009; Lopez et al., 2008).

Um das Programmierkonzept «Variablen» zu veranschaulichen, können Analogien verwendet werden (Kohn & Komm, 2019; Stechert, 2020b). Beispielsweise werden diese häufig als Box erklärt, in welcher ein konkreter Wert gelegt werden kann (Van Der Werf et al., 2023). Die Verwendung solcher Analogien wird auch kritisch betrachtet (Sorva, 2018). Eine Studie zeigte, dass diese Box-Analogie das Verständnis von Variablen zunächst erleichtern, jedoch bei komplexeren Aufgabenstellungen zu Verwirrung führen kann (Hermans et al., 2018). Gerade für junge Lernende gibt es noch keinen wissenschaftlichen Konsens darüber, wie und wann das Programmierkonzept «Variablen» im Unterricht thematisiert werden soll.

Ziel dieser Arbeit ist es somit den Informatikunterricht in Bezug auf das Programmierkonzept «Variablen» näher zu beleuchten und zu ermitteln, welche Analogien sich für Lernende im Alter von 10–12 Jahren geeignet sein könnten. Entlang des DBR-Prozesses wird das Programmierkonzept «Variablen» zunächst im fachwissenschaftlichen und fachdidaktischen Kontext beleuchtet und ehe bereits bestehende Ansätze zur Vermittlung des Programmierkonzepts

«Variablen» aufgezeigt werden (Kapitel 2). Daran schliesst die Darstellung Methodik dieser Arbeit (Kapitel 3). Um ein tieferes Verständnis über den aktuellen Forschungsstand zum Programmierkonzept «Variablen» und zur Verwendung von Analogien zu erlangen, werden bestehende Lehrmittel und Forschungsbeiträge untersucht (Kapitel 4) und die Sichtweise von Experten und Expertinnen aus der Fachdidaktik und Praxis eingeholt (Kapitel 5). Die Erkenntnisse aus diesen Untersuchungen bildeten die Grundlage für eine Befragung von Lernenden, mittels derer sowohl potentielle Alltagsanalogien als auch Vorstellungen der Lernenden über Programmierkonzept «Variablen» untersucht wurden (Kapitel 6). In der Diskussion werden die Ergebnisse der drei Untersuchungen zusammengefasst, im Hinblick auf die Zielsetzung dieser Arbeit interpretiert (Kapitel 7). Abschliessend wird ein Fazit gezogen (Kapitel 8)

2. Theorie und Forschungsstand

In diesem Kapitel soll ein Überblick über den aktuellen Forschungsstand zum Programmieren mit Variablen vermittelt werden. In Kapitel 2.1 werden fachwissenschaftliche Grundlagen der Informatik und die Eigenschaften des Programmierkonzepts «Variablen» beschrieben. In Kapitel 2.2 folgen fachdidaktische Grundlagen zum Programmieren, wobei auch eine Definition der didaktischen Methode «Analogie» erfolgt. Zuletzt werden in Kapitel 2.3 relevante wissenschaftliche Erkenntnisse zur Vermittlung des Programmierkonzepts «Variable» thematisiert.

2.1 Fachwissenschaftliche Grundlagen der Informatik

In diesem Kapitel werden die grundlegenden Konzepte und Prinzipien der Informatik erläutert, die für diese Arbeit von Relevanz sind. Es wird kurz definiert, was unter dem Begriff «Programmieren» verstanden wird und worum es sich bei Programmiersprachen bzw. Programmierumgebungen handelt. Darüber hinaus werden wesentliche Programmierkonzepte thematisiert. Das Programmierkonzept «Variablen» wird anschliessend näher beleuchtet und von anderen Fachbereichen abgegrenzt.

2.1.1 Programmieren

Das Programmieren umfasst den gesamten Prozess des Schreibens von Programmen (Guzdial, 2016, S. 2). Dieser Prozess beinhaltet den Entwurf, die Implementierung und das Testen von Computerprogrammen (Hielscher & Döbeli Honegger, 2015). Programme können in unterschiedlichen Formen vorliegen, beispielsweise textbasiert oder blockbasiert, und stellen einen Rechenprozess dar, der vom Computer ausgeführt wird (Guzdial, 2016, S. 2).

2.1.2 Programmiersprache und Programmierumgebung

Um die Entwicklung komplexer Programme zu ermöglichen, wurden unterschiedliche Programmiersprachen entwickelt, die einem Pseudocode ähneln sowie sprachlich leicht verständlich und in Maschinencode umsetzbar sind (Brookshear, 2012, S. 240). Bekannt sind beispielsweise C++, Java oder Python (Hielscher & Döbeli Honegger, 2015). Programme werden meist in sogenannten Entwicklungsumgebungen erstellt, die unter anderem Editor, Compiler und Debugging-Tools beinhalten (Brookshear, 2012, S. 302). Für Programmieranfänger sind solche Programmierumgebungen jedoch aufgrund ihrer Komplexität nicht geeignet. Aus diesem Grund wurden Programmierlernumgebungen entwickelt. Diese Umgebungen sind durch eine altersentsprechende Darstellung sowie spielerische Ansätze gekennzeichnet und dienen dem Ziel, Erfolgserlebnisse für die Lernenden zu schaffen und ihnen Raum zu geben, um ihre eigenen Ideen umzusetzen. Genutzt werden dafür vereinfachte Programmiersprachen, unterstützende Visualisierungen und verschiedene Feedbackformen. (Hielscher & Döbeli

Honegger, 2015). In der Literatur wird dabei oft zwischen textbasierten Programmiersprachen wie XLogo¹ und blockbasierten Programmiersprachen wie Scratch² unterschieden. Zweitere ermöglichen eine visuelle Darstellung von Befehlen innerhalb eines Programms. Dies bietet für die Lernenden den Vorteil, dass die Programmiersprache weniger fehleranfällig ist und keine komplexe Syntax aufweist (Kalogiannakis & Papadakis, 2018). Studierende schätzen die Übersicht der Befehle und das Drag-and-Drop-Prinzip³, bemerken jedoch einen Verlust von Authentizität und Überlegenheit gegenüber textbasierten Programmiersprachen (Weintrop & Wilensky, 2015).

2.1.3 Programmierkonzepte

Informatiker:innen bedienen sich beim Programmieren verschiedener Konzepte. Die sieben zu erlernenden Konzepte wurden durch Brennan und Resnick (2012) als «computational thinking concepts» bezeichnet und können Tabelle 1 entnommen werden. Variablen werden in diesem Kontext dem Konzept der Datenstrukturen zugeordnet.

Tabelle 1

Beschreibung der sieben relevanten Programmierkonzepte.

Konzept	Beschreibung
Sequenzen	Erstellen einer Reihe von Einzelschritten oder Anweisungen, die von einem Computer ausgeführt werden können
Schleifen	Mechanismen zum mehrmaligen Ausführen derselben Sequenz
Ereignisse	Ein Ereignis löst ein anderes aus
Parallelität	Anweisungen, die gleichzeitig ausgeführt werden
Bedingungen	Entscheidungen aufgrund von Voraussetzungen treffen
Operatoren	Unterstützung von mathematischen und logischen Ausdrücken, z. B. mathematische Grundoperationen, Sinusfunktion
Datenstrukturen	Ermöglichen das Speichern, Abrufen und Aktualisieren von Werten; dafür können Variablen oder Listen verwendet werden

Anmerkung. Darstellung aus eigenem Leistungsnachweis (Gisler & Regli, 2023) in Anlehnung an Brennan & Resnick (2012) und Ching et al. (2018)

¹ Programmierumgebung XLogo: <https://xlogo.inf.ethz.ch/release/latest/#/>

² Programmierumgebung Scratch: <https://scratch.mit.edu/>

³ Möglichkeit die Befehle mit der Maus hin und her zu ziehen

Im Lehrplan 21 (D-EDK, 2016a) sind im Modullehrplan Medien und Informatik Bezüge zu diesen Programmierkonzepten zu finden. In der Kompetenz MI2.2 werden namentlich Schleifen, bedingte Anweisungen (Bedingungen) sowie Parameter für Zyklus 2 genannt. Variablen sind in der Kompetenzbeschreibung zu Zyklus 3 zu finden.

2.1.4 Programmierkonzept «Variablen»

Das Wort «Variable» stammt aus dem Lateinischen und bedeutet wörtlich übersetzt «Veränderliche» (Fischer & Hofer, 2011, S. 956). Genauer handelt es sich dabei um «einen Bezeichner mit einem bestimmten Datentyp und sich erst im Programmablauf ergebendem, veränderlichem Wert» (Fischer & Hofer, 2011, S. 956). Bagge und Grover (2020) geben auch an, dass Variablen in der Informatik Einheiten darstellen, deren Werte sich verändern können. Diese Werte werden in einem bestimmten Bereich im Arbeitsspeicher des Computers gespeichert (Bagge & Grover, 2020). Variablen können verwendet werden, um eine generalisierende Lösung eines Codes aufzuzeigen, Code zu vereinfachen und gewisse Teile zu verstecken (Bagge & Grover, 2020).

Eine Variable hat einen Namen, unter welchem diese angesprochen und ihr Wert verändert werden kann (Claus & Schwill, 2006, S. 718). Eine Variable verfügt somit über zwei Teile. Sie besteht aus einem Namen, der verwendet wird, um in einem Algorithmus oder Programm darauf zu verweisen, und aus einem Wert, der im Arbeitsspeicher gespeichert wird. Werte, die in Variablen gespeichert werden, gehören zu einem Datentyp. Wie Tabelle 2 zeigt, sind unterschiedliche Datentypen bekannt. Die Unterscheidung des Datentyps ist entscheidend, da in Abhängigkeit vom Datentyp unterschiedliche Operationen auf den Variablen laufen können (Bagge & Grover, 2020). Beispielsweise können auf numerischen Variablen traditionelle arithmetische Operationen, wie «addieren» oder «subtrahieren», aber auch vergleichende Operationen, wie «grösser als», erfolgen (Brookshear, 2012, S. 250).

Tabelle 2

Beschreibung der häufigsten Datentypen in der Informatik mit Beispielen

Datentyp	Beschreibung	Beispiel
Integer	Zahlenwert ohne Dezimalstelle	1, 35, 693
Float	Zahlenwerte mit Dezimalstelle	2,45, 69,1, 0,496
String	Text oder eine Kette von Zeichen	«Hallo Welt»
Boolean	Wahrheitswerte	true / false

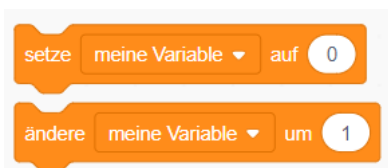
Anmerkung. Eigene Darstellung in Anlehnung an Bagge und Grover (2020)

Variablen weisen meistens einen logischen Wertebereich auf. Beispielsweise nimmt eine Variable mit dem Namen «Alter» Werte von 1–100 an und sollte eine ganze Zahl darstellen. Eine negative Zahl ist bei dieser Variable unwahrscheinlich. Das Verständnis vom logischen Bereich der Variable kann bei der Aufstellung von Validierungsregeln helfen (Bagge & Grover, 2020).

Die Verlinkung von einem Wert zu einer Variable wird als «Zuweisung» (engl. Assignment) bezeichnet. Dabei wird der angegebene Wert in dem durch die Variable identifizierten Speicherbereich abgespeichert (Brookshear, 2012, S. 254–255). In Scratch können dafür die in Abbildung 1 dargestellten Blöcke verwendet werden. In textbasierten Programmiersprachen wird die Zuweisung oft mit dem Gleichzeichen umgesetzt. In diesem Kontext bedeutet das Zeichen jedoch nicht «ist gleich». Die Zuweisung erfolgt stets von rechts nach links. Es ist möglich, dass sich der bisherige Wert über eine mathematische Operation⁴ durch den neuen Wert ändert (Bagge & Grover, 2020, S. 231).

Abbildung 1

Befehle zum Verändern von Variablen in Scratch



Anmerkung. Bildschirmaufnahme aus Scratch.

Wie bereits erklärt, haben Variablen einen Namen. Dieser Name kann von der programmierenden Person gewählt werden. Es gilt, den Namen so zu wählen, dass möglichst klar ist, was für eine Bedeutung dem Wert hinter der Variable zugewiesen werden kann. Bei Variablen mit mehreren Wörtern wird meistens die Pascal Case Convention angewendet. (Bagge & Grover, 2020) Dabei werden alle Anfangsbuchstaben der Wörter grossgeschrieben und auf Abstände zwischen den Wörtern verzichtet (McKenzie, 2021).

Eine Variable wird beim Start eines Programms initialisiert. Dabei sollte ihr anfängliche Wert zugewiesen werden (Bagge & Grover, 2020, S. 232). Dazu muss die programmierende Person entscheiden, welchen Wert die Variable beim Start des Programms haben sollte. In Scratch beispielsweise wird bei einer Wiederholung des Programms mit dem letzten Wert der Variable weitergearbeitet. Deshalb ist es entscheidend, bei der Erstellung eines Programms den gewünschten anfänglichen Wert zu initialisieren.

⁴ Beispielsweise addieren, subtrahieren, multiplizieren oder dividieren

Variablen können sich innerhalb eines Programms verändern, aber auch den gleichen Wert behalten. Verändern sie sich beim Durchlaufen des Programms, wird auch vom Updaten von Variablen gesprochen. Eine Variable speichert jedoch nur den aktuellen Wert. In der Programmierungsumgebung Scratch existieren verschiedene Möglichkeiten, um den Wert einer Variable zu verändern (Abbildung 1). Variablen können lokalen oder globalen Charakter haben. Eine Variable, die nur innerhalb von einer Funktion (in Scratch eine Figur) verwendet werden kann, ist lokal. Eine Variable, die von überall (in Scratch von allen Figuren) verändert werden kann, ist global (Brookshear, 2012, S. 261).

Eine Variable kann, wie dieses Kapitel zeigt, durch unterschiedliche Eigenschaften charakterisiert werden. In Tabelle 3 sind diese Eigenschaften von Variablen zusammengefasst dargestellt.

Tabelle 3

Zusammenfassung relevante Eigenschaften des Programmierkonzepts «Variablen»

Eigenschaft	Erklärung
Veränderbarkeit/updaten	Der Wert einer Variable kann sich im Laufe des Programms verändern. Die Variable speichert jeweils nur den aktuellsten Wert.
Benennung	Variablen haben einen Namen, unter dem sie angesprochen werden.
Datentypen	Variablen können unterschiedliche Datentypen speichern.
Logischer Wertebereich	Variablen haben oft einen logischen Wertebereich, den der Wert der Variable einnehmen kann.
Zuweisung	Bei der Zuweisung wird ein Wert einer Variable zugewiesen.
Initialisierung	Zu Beginn eines Programms wird einer Variable ein Anfangswert zugewiesen.
Zugriffsmöglichkeit	Auf eine Variable kann lokal oder global zugegriffen werden.

Anmerkung. Eigene Darstellung

2.1.5 Abgrenzung zu anderen Fachbereichen

Der Begriff «Variable» wird nicht nur im Bereich der Informatik verwendet, sondern findet sich in unterschiedlichen Fachwissenschaften wieder. In der Mathematik gilt eine Variable oft als unbekannte Grösse oder Platzhalter für Zahlen oder andere mathematische Objekte (Ely & Adams, 2012). In der Statistik ist eine Variable ein Merkmal oder eine Eigenschaft, die bei den Elementen einer statistischen Untersuchungseinheit gemessen werden kann (Bevans, 2022). Auch in der Physik, der Wirtschaft oder auch der Biologie wird der Begriff «Variable» unter den spezifischen Anforderungen des Fachs verwendet. Das Verständnis des Programmierkonzepts «Variable» kann durch diese unterschiedlichen Definitionen erschwert werden, was in Kapitel 2.3.2 anhand des Fachs Mathematik illustriert wird.

2.2 Fachdidaktische Grundlagen des Programmierens

Grover und Pea (2018) zählen informatische Kompetenzen zu den fundamentalen Fähigkeiten, um in der heutigen Gesellschaft zu agieren. Deshalb gilt es zu ergründen, wie diese Kompetenzen im schulischen Kontext gefördert werden können. In diesem Kapitel werden allgemeine fachdidaktische Erkenntnisse präsentiert, die für die vorliegende Arbeit als relevant betrachtet werden.

2.2.1 Computational Thinking

Computational Thinking (CT) gilt als relevante Kompetenzbeschreibung im informatischen Bereich. Wing (2006) definiert CT als Denkweise von Informatiker:innen. Zu diesem Prozess gehören die Problemformulierung und die Erarbeitung von Lösungsvorschlägen, die sowohl von Menschen als auch von einer Maschine umgesetzt werden können (Wing, 2006). Dies erfordert logisches und algorithmisches Denken, Mustererkennung, Abstraktion und Generalisierung von Problemen, Evaluation und Automation (Grover & Pea, 2018). Zu verstehen, wie ein Computer funktioniert, und zu begreifen, dass dieser nicht unbedingt intelligent ist, stufen auch Kohn und Komm (2019) im heutigen Zeitalter als bedeutend ein. Programmieren wird als relevanter Bestandteil der Informatik und damit von CT betrachtet (Grover & Pea, 2013).

2.2.2 Vermittlung von Programmierkompetenzen

Das Erlernen des Programmierens ist für Lernende anspruchsvoll (Havenga et al., 2013). Schwierigkeiten bereiten ihnen hauptsächlich die eigenständige Entwicklung von Programmen (Macrides et al., 2022; Yusoff et al., 2020), das Kombinieren von Befehlen (Robins et al., 2003) und das Lesen von Algorithmen (Medeiros et al., 2019). Eine weitere Herausforderung stellt das Verständnis von Programmierkonzepten wie Schleife oder Variablen dar (Grover et al.,

2019). Aus diesem Grund wurde in der Fachdidaktik nach Methoden gesucht, um diesen Lernprozess zu unterstützen.

Einerseits gilt es als hilfreich, wenn neue Inhalte schrittweise eingeführt werden (Salac et al., 2021). Die verschiedenen Programmierkonzepte werden deshalb oft nacheinander behandelt (Rich et al., 2018; Wiegand et al., 2016). Die Programmierkonzepte können in Programmierlernumgebungen (Kapitel 2.1.2), mit Lernrobotern oder auch durch sogenannte Unplugged-Aktivitäten erarbeitet werden. Diverse Studien belegen einen Zuwachs der CT-Kompetenzen beim Einsatz selbiger (Brackmann et al., 2017; del Olmo-Muñoz et al., 2020; Sun et al., 2021). Munasinghe, Bell und Robins (2023) beschreiben Unplugged-Aktivitäten sogar als Katalysator im Informatikunterricht. Entscheidend ist dabei, dass eine Verlinkung zwischen den Aktivitäten und dem Programmierkonzept sowie der damit verbundenen Fachsprache hergestellt wird (Waite et al., 2019). Gemäss der Theorie der «Semantic Wave» (Maton, 2013) ist es sogar zielführend, mit der korrekten Terminologie des Konzepts zu beginnen und anschliessend eine altersstufenentsprechende Sprache zu verwenden. In diesem Schritt können Analogien und Unplugged-Aktivitäten den Lernprozess stark unterstützen (Curzon et al., 2018). Auf diese Weise soll den Lernenden dabei geholfen werden, das neue Konzept zu erschliessen und es in einen Zusammenhang mit Bekanntem zu setzen (Waite et al., 2019). Das neue Wissen soll auf ein abstrakteres Niveau gebracht werden, indem zwischen der Vereinfachung und dem Konzept unterschieden wird.

Ein anderer Ansatz in der Programmierdidaktik ist der Einsatz von sogenannten Worked Examples, bei dem Lernende aktiv konkrete Beispiele für die Lösung eines Problems analysieren. (Leppink et al., 2014). Im Bereich des Programmierens werden dafür häufig Codebeispiele genauer betrachtet (Margulieux et al., 2020). Didaktisiert werden diese Worked Examples im Ansatz «Use-Modify-Create». Die Lernenden sollen dabei zuerst mit einem vorgegebenen Programm interagieren. Handelt es sich beispielsweise bei dem Programm um ein Spiel, wird dieses im Schritt «Use» gespielt. Im Schritt «Modify» verändern sie das vorgegebene Programm. Diese Modifikationen können auf unterschiedlichen Schwierigkeitsstufen erfolgen, fördern das Verständnis des Programms und ermöglichen die Entwicklung neuer Fähigkeiten. In der letzten Phase werden diese Fähigkeiten in Projekten mit eigenen Ideen angewendet (Lee et al., 2011). Lytle et al. (2019) stellten fest, dass Lernende mit UMC einen natürlichen Lernprozess verfolgen und mehr Verantwortung in der Kreation von eigenen Projekten erfahren.

2.2.3 Didaktische Rekonstruktion

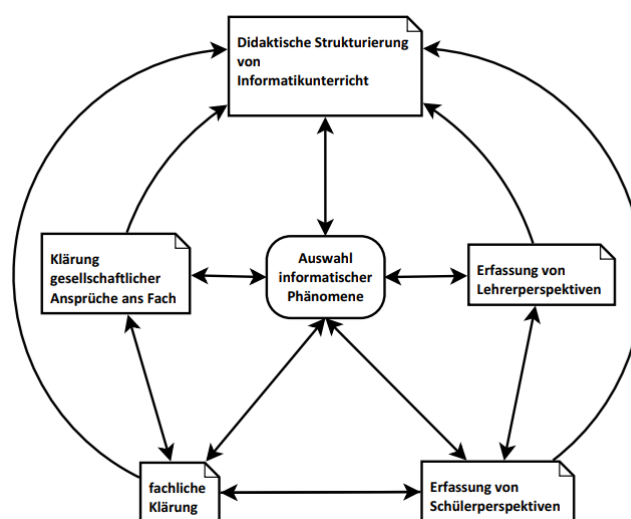
Bei der Planung und Erstellung von Unterrichtsmaterial sollten neben dem fachlichen Inhalt auch die Ziele und die Perspektive der Lernenden in Betracht gezogen werden. All diese Aspekte vereint das Modell der Didaktischen Rekonstruktion (Kattmann et al., 1997). Für den

Informatikunterricht wurde dieses Modell erweitert. Diethelm, Dröge, Mesaros und Dunnebie (2011) schlagen das in Abbildung 2 gezeigte Modell vor. In diesem wird die Auswahl von informatischen Phänomenen in den Mittelpunkt gerückt. Diethelm und Dörge (2011, S. 9) definieren ein informatisches Phänomen als «ein Ereignis, das durch automatisierte Informationsverarbeitung verursacht wird und im realen oder mentalen Handlungsumfeld der Schüler stattfindet.» Bei der fachlichen Klärung wird der zu vermittelnde Inhalt betrachtet und abgeklärt, in welcher Tiefe das Thema unterrichtet wird. Dafür muss angeschaut werden, inwiefern die verwendete Erklärung passend ist (Diethelm et al., 2011). Diese Erklärungen sollen nicht nur fachlich korrekt sondern auch verständlich für Lernende sein. Beim Erlernen von Programmierkompetenzen nimmt das Vorwissen der Lernenden eine zentrale Rolle ein (Meyer-Baron, 2024). Das zeigt sich dadurch, dass Lernende auf ihre Vorerfahrungen zurückgreifen, wenn sie programmieren. (Rich et al., 2018). Speziell im Fach Informatik ist es entscheidend, die Perspektive der Lehrperson zu berücksichtigen, da das Fachwissen der Lehrpersonen stark variieren kann (Hubwieser et al., 2011). Bei der Aufarbeitung und Gestaltung von Unterrichtsmaterial muss berücksichtigt werden, welche Vorstellungen Lehrpersonen von dem im Zentrum stehenden Konzept haben (Diethelm et al., 2012). Die gesellschaftliche Perspektive und den damit eingehenden Beitrag zur Allgemeinbildung nehmen in diesem Modell ebenfalls Einfluss (Diethelm et al., 2011).

Das verbindende Glied bildet die didaktische Strukturierung, das als Resultat dieses wechselwirkenden und iterativen Prozesses gilt. Die wechselwirkenden Einflüsse der beschriebenen Aspekte sind in Abbildung 2 visualisiert.

Abbildung 2

Modell der didaktische Rekonstruktion für den Informatikunterricht



Anmerkung. Abbildung aus Diethelm et al. (2011)

2.2.4 Vorstellungen von Lernenden

Im Folgenden wird besagte Perspektive der Lernenden genauer in den Blick genommen. Vorstellungen von Lernenden werden auch als «mentale Modelle» bezeichnet. Sie werden definiert als innere Repräsentationen über ein Konzept, die sich durch Erfahrung entwickeln haben (Babari et al., 2023). In der Studie über Internetvorstellungen wird der Begriff «Vorstellung» verwendet, um das gesamthafte Verständnis sowie die Überzeugungen und Ideen einer Person zu einem Konzept zu beschreiben. Laut Diethelm et al. (2012) ist es dabei entscheidend, alle Arten von Vorstellungen, ob fehlerhaft oder nicht, zu respektieren. Deshalb unterscheiden Babari et al. (2023) zwischen intuitiven Vorstellungen, elaborierten Vorstellungen und Fehlvorstellungen (FV). Die Beschreibung dieser drei Kategorien ist in Tabelle 4 zusammengefasst.

Tabelle 4

Beschreibung der drei Arten von Vorstellungen

Art der Vorstellung	Beschreibung	Quelle
intuitive Vorstellung	einfache Verallgemeinerungen, die im Alltag über soziale Interaktion entwickelt werden	(Linn & Songer, 1991)
elaborierte Vorstellung	Vorstellungen, die Elemente des fachwissenschaftlichen Denkens beinhalten und ein vertieftes Verständnis aufzeigen	(Babari et al., 2023)
Fehlvorstellung	Vorstellungen, die von der fachwissenschaftlichen Definition abweichen	(Edelsbrunner et al., 2018)

Anmerkung. Eigene Darstellung in Anlehnung an Babari et al. (2023)

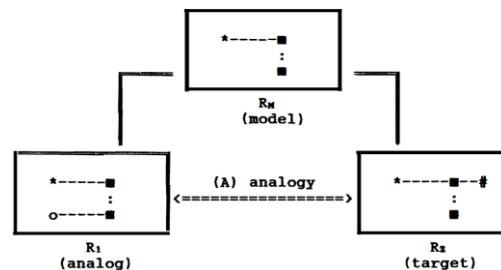
2.2.5 Analogie in der Didaktik

Wie in Kapitel 2.2.2 beschrieben, können unter anderem Analogien verwendet werden, um anspruchsvolle Konzepte zu erklären. Der Begriff «Analogie» stammt aus dem Griechischen und bedeutet übersetzt «Ähnlichkeit» (Haider, 2010). Analogien stellen somit Ähnlichkeiten oder auch Parallelen zu einem Lerninhalt her (Haider & Fölling-Albers, 2020). Analogien werden bereits seit vielen Jahren in den Naturwissenschaften in der Lehre eingesetzt (Wieczorek et al., 2017). Duit (1991) hat in seiner Arbeit eine Definition für Analogien auf einer formalen Ebene gegeben. Abbildung 3 zeigt die möglichen Repräsentationsformen R1, R2 und RM. R1 und R2 weisen identische Merkmale auf, die mit einem Modell (RM) dargestellt werden können. In diesem Fall wird von einem Analogiemodell gesprochen (Haider, 2010). Duit (1991) hält fest, dass es in einer solchen analogen Beziehung einer Symmetrie in Bezug auf die Struktur des zu erklärenden Konzepts bedarf. Das zu erklärende Objekt wird in der Literatur meist «Ziel» (engl. target) genannt. Die Analogie wird dabei teilweise auch als «Quelle» oder «Basis»

bezeichnet, weil sie im Lehr- und Lernprozess als solche fungiert (Glynn et al., 1989; Rumelhart & Norman, 1981). In der vorliegenden Arbeit werden das Zielkonzept «Variablen» und die entsprechenden Analogien thematisiert.

Abbildung 3

Darstellung der formalen Definition einer Analogie



Anmerkung. Darstellung aus Duit (1991).

Neben dem Begriff «Analogie» wird teilweise auch das Wort «Metapher» verwendet (Aubusson et al., 2006). Aubusson et al. (2006) beschreiben die Unterschiede zwischen diesen beiden Begriffen. Während Metaphern eher als «A is said to be B» (Aubusson et al., 2006, S. 2) beschrieben werden, können Analogien als «A is like B» versinnbildlicht werden (Aubusson et al., 2006, S. 2). Analogien haben somit bestimmte Eigenschaften mit dem Zielkonzept gemeinsam. Sie können sich jedoch vom Zielkonzept in gewissen Punkten unterscheiden. So sollen diese Ähnlichkeiten und Unterschiede in Analogien explizit ausgedrückt werden; in Metaphern sind sie verdeckt (Aubusson et al., 2006).

Neben der formalen Beschreibung von Analogien durch Duit (1991) finden sich in der Literatur auch praxisorientiertere Bezüge. Analogien werden herangezogen, um Unterrichtsgegenstände zu visualisieren, die zu komplex und abstrakt sind (Haider & Fölling-Albers, 2020). Lernende nutzen ihr Wissen aus einem sekundären und verständlichen Bereich, um Verbindungen zum zu erlernenden primären Konzept herzustellen (Glynn & Duit, 1995; Kircher, 1995). In diesem Sinne erfüllen Analogien eine Brücken- (Glynn & Duit, 1995) oder Mittlerfunktion (Kircher, 1995). Aus lernpsychologischer Sicht spielt bei den Analogien die Structure-Mapping-Theorie von Gentner (1983) eine Rolle, die ebendiesen Prozess des Nutzens von bekanntem Wissen für das Verständnis eines neuen Lernbereichs beschreibt.

Damit Analogien genutzt werden können, müssen sie gewisse Merkmale aufweisen. Kircher und Girwidz (2020) haben notwendige Bedingungen für Analogien im Physikunterricht gesammelt:

- Vertrautheit: Die Analogie muss sowohl oberflächenähnlich (Gentner, 1989) als auch den Lernenden vertraut sein.
- Partielle Isomorphie: Die Analogie muss eine Tiefenstrukturähnlichkeit aufzeigen. Das bedeutet, dass gewisse Teile der beiden Lernbereiche Gleichheiten aufzeigen müssen.
- Eigengesetzlichkeit der Lernbereiche: Die Analogie kann unterschiedliche Eigenschaften aufweisen, die jedoch zwingend thematisiert werden sollen.

Um über die Verwendung einer Analogie zu entscheiden, müssen diese und das Zielkonzept demnach genau untersucht werden. So hält Aubusson (2006) fest, dass Analogien das Verständnis nur fördern können, wenn sie umfassend analysiert werden und irreführende Merkmale erkannt sowie angesprochen werden. Die Einführung und Umsetzung von Analogien ist anspruchsvoll (Harrison & Treagust, 1993), weshalb der *«Focus-Action-Reflection»*-Leitfaden eine systematische Einführung für Analogien vorgibt (Treagust et al., 1998) und in Tabelle 5 dargestellt wird. Unterricht, der nach diesem Leitfaden aufgebaut ist, kann das Verständnis von Lernenden über das zu erlernende Konstrukt steigern (Harrison & Treagust, 2000).

Tabelle 5

«Focus -Action-Reflection»-Leitfaden für die Nutzung von Analogien im Unterricht

Fokus (vor dem Unterricht)	
Konzept	Ist das Konzept anspruchsvoll, unklar oder abstrakt?
Lernende	Welche Vorstellungen haben die Lernenden bereits über das Konzept?
Erfahrung	Welche vertrauten Erfahrungen, die genutzt werden können, haben die Lernenden?
Aktion (während des Unterrichts)	
Gemeinsamkeiten	Überprüfen, ob die Lernenden mit der Analogie vertraut sind Über die Ähnlichkeiten zwischen Analogie und Konzept sprechen und oberflächliche sowie tiefere Gemeinsamkeiten ergründen
Unterschiede	Über Unterschiede zwischen Analogie und Konzept sprechen
Reflektion (nach dem Unterricht)	
Schlussfolgerung	War die Analogie verständlich, nützlich oder verwirrend?
Verbesserung	Welche Veränderung ist für die nächste Lektion erforderlich? Welche Veränderung ist erforderlich, wenn die Analogie das nächste Mal verwendet wird?

Anmerkung. Eigene Darstellung in Anlehnung an Treagus et al. (1998)

Neben der Reflektion über Ähnlichkeiten und Abweichungen ist es ebenfalls von Bedeutung, zu begründen, weshalb eine Analogie verwendet wird. Curtis und Reigeluth (1984) haben in ihrer Untersuchung von Lehrmitteln drei unterschiedliche Arten von Analogien identifiziert: die einfache Analogie⁵, bei der der Vergleich der beiden Konzepte genannt wird, die angereicherte Analogie⁶ als erklärende Analogie, bei der zusätzlich eine Begründung beschrieben wird, und die erweiterte Analogie⁷, bei der alle oder einige Teilaspekte begründet sind, die also mehrere gemeinsame Attribute enthält, welche begründet werden.

Analogien werden in unterschiedlichen Fachbereichen verwendet, um ein abstraktes Konzept erklärbar zu machen, Im nächsten Kapitel wird ergründet, inwiefern dies auch für die Didaktik der Informatik gilt.

2.2.6 Analogien in der Informatik

Zahlreiche Konzepte im informatischen Bereich sind abstrakt und nicht leicht zugänglich (Saxena et al., 2023). Deswegen werden auch hier Analogien (Fincher et al., 2020) und Notional Machines (Van Der Werf et al., 2023) genutzt. Hermans (2019) beschreibt Notional Machines als « didaktisches Mittel, um die Ausführung eines Programms auf einer realen Maschine zu verstehen »⁸. Diese Vereinfachungen sollen die zentralen Aspekte eines Konzepts beleuchten (Van Der Werf et al., 2023) und damit unsichtbare Vorgänge sichtbar machen, die für den Kompetenzzuwachs von jungen Lernenden relevant sind (Fincher et al., 2020). Analogien können als mögliche Formen oder Teilaspekte von Notional Machines verstanden werden (Fincher et al., 2020; Hermans, 2019; Stechert, 2020b⁹). Weitere mögliche Formen von Notional Machines sind Repräsentationen, Regeln und Aktionen (Hermans, 2019; Stechert, 2020c). Fincher et al. (2020) stellten fest, dass im Unterricht verwendete Notional Machines oft auf einer Analogie aufbauen. Die Verwendung von Analogien oder Notional Machines wird jedoch auch kritisch betrachtet, da sie durch die Vereinfachung gewisser Vorgänge auch Ursprünge von FV beim Programmieren sein können (Du Boulay, 1986).

2.2.7 Zusammenfassung

Im Informatikunterricht stoßen Lernende auf Herausforderungen wie die eigenständige Entwicklung von Programmen und das Verständnis grundlegender Konzepte wie Variablen. Unterstützende Methoden umfassen die schrittweise Einführung neuer Inhalte, die Vereinfachung

⁵ Engl. simple analogy

⁶ Engl. enriched analogy

⁷ Engl. extended analogy

⁸ Übersetzung aus dem Englischen. Ursprüngliches Zitat: «A notional machines is a didactical tool that helps a learner understand the workings of the source code on an actual machine»

⁹ In Hermans (2019) und Stechert (2020c) wird von Metaphern gesprochen, diese können jedoch wie in Kapitel 2.2.5 erläutert als Analogien verstanden werden.

und Visualisierung in Unplugged-Aktivitäten und das «Use-Modify-Create»-Modell (Kapitel 2.2.2). Das Planungsmodell «Didaktische Rekonstruktion» zeigt auf, dass im Informatikunterricht nicht nur fachliche Aspekte berücksichtigt werden müssen, sondern auch Perspektiven der Lernenden und Lehrenden sowie die Bedeutung für die Gesellschaft Einfluss haben (Kapitel 2.2.3). Vorstellungen von Lernenden können intuitiv, elaboriert oder falsch sein (Kapitel 2.2.4). Analogien erleichtern das Verständnis komplexer Konzepte, indem sie es ermöglichen, Verbindungen zu vertrauten Bereichen herzustellen (Kapitel 2.2.5). In der Informatikdidaktik werden sie als Teil von Notional Machines verwendet, um abstrakte Vorgänge zu veranschaulichen (Kapitel 2.2.6).

2.3 Wissenschaftliche Erkenntnisse zur Vermittlung von Variablen

Das Programmierkonzept «Variablen» gilt im Bereich der Informatik als bedeutend und anspruchsvoll zugleich (Grover et al., 2019; Hermans et al., 2018; Kohn, 2017). Lernende mit einem kompetenten Verständnis dieses Konzepts können Code besser lesen und verstehen (Corney et al., 2011; Lister et al., 2009; Lopez et al., 2008). Schwierigkeiten beim Verstehen vom Programmierkonzept «Variablen» gelten sowohl für block- als auch textbasierte Programmiersprachen (Grover & Basu, 2017). Die Untersuchung von Scratchprojekten zeigte, dass Variablen von Lernenden (Maloney et al., 2008; Seiter & Foreman, 2013) wie auch im Allgemeinen (Meerbaum-Salant et al., 2013) selten verwendet werden. Aus diesem Grund wird eine explizite Einführung von Variablen gefordert (Rich et al., 2020). In diesem Kapitel wird aufgezeigt, welche bereits bekannten Forschungsergebnisse für die Entwicklung eines solchen Unterrichtssettings zum Thema «Variablen» als relevant betrachtet werden.

2.3.1 Fehlvorstellungen zu Variablen

Die Perspektive der Lernenden nimmt wie in Kapitel 2.2.3 thematisiert eine zentrale Rolle in der Unterrichtsplanung ein. FV können dabei helfen, das Erlernen von Programmierkonzepten besser zu verstehen (Sorva, 2018). Sorva (2018) hat bei Novizen solche FV im Bereich des Programmierens gesammelt. Diese wurden von Hartman et al. (2022) studiert, mit FV aus anderen Studien verglichen und Programmierkonzepten zugeordnet. FV, welche sich auf Variablen beziehen sind in Tabelle 6 ersichtlich und werden im folgenden genauer erläutert.

Tabelle 6

Zusammenstellung bekannter Fehlvorstellungen von Lernenden über das Programmierkonzept «Variablen»

Nr.	Fehlvorstellungen	Quelle ¹⁰
1	Eine Variable kann mehrere Werte auf einmal enthalten/merkt sich alte Werte.	Nr. 1 (Sorva, 2018)
2	Die Variable enthält den ersten zugewiesenen Wert.	(Žanko et al., 2019)
3	Die Variable ist die Summe aller zuvor zugewiesenen Werte.	(Žanko et al., 2019)
4	Eine Variable ist (nur) eine Verknüpfung eines Namens mit einem veränderbaren Wert, die nicht im Computer gespeichert wird.	Nr. 3 (Sorva, 2018)
5	Primitive Zuweisung wie $a = b+1$ speichert Gleichungen oder nicht aufgelöste Ausdrücke.	Nr. 4 (Kohn & Komm, 2019; Sorva, 2018)
6	Bei einer Zuweisung wie $a = b$ werden die Werte von einer Variable auf die andere verschoben; a ist anschliessend leer.	Nr. 11 (Sorva, 2018)
7	Die Inkrementierung einer Zählervariablen ist eine unteilbare Operation (keine separate Auswertung der rechten Seite).	Nr. 13 (Sorva, 2018)
8	Die primitive Zuordnung funktioniert in umgekehrter Richtung. $10 = a$	Nr. 11 (Kohn & Komm, 2019; Sorva, 2018)
9	Die primitive Zuordnung funktioniert in beide Richtungen (Tausch).	Nr. 7 (Sorva, 2018)

Anmerkung. Eigene Darstellung in Anlehnung an Hartmann (persönliche Kommunikation, 21. September 2023)

Die Ursprünge dieser FV werden einerseits darin gesehen, dass Lernende mathematische Gesetzmässigkeiten anwenden (Kapitel 2.3.2) oder Analogien falsch aufgefasst haben (Stechert, 2020b; Kapitel 2.3.3). Manchmal fokussieren Lernende sich mehr auf den Zweck des Programmcodes und haben dadurch Schwierigkeiten beim Verständnis der Struktur. Das zeigt sich in FV 7. Lernende verstehen, dass die Variable a um eins erhöht wird. Was dabei oft nicht angeschaut wird, ist, dass zuerst der Ausdruck ausgewertet und dann der neue Wert der Variable zugewiesen wird (Sorva, 2018). FV 2 und 3 weisen Bezüge zum Verändern und

¹⁰ Die Nummerierung bezieht sich auf die Zusammenstellung von Sorva (2018).

Aktualisieren von Variablen auf. Luo et al. (2022) stellten auch fest, dass Lernende Schwierigkeiten hatten, Initialisierung und Aktualisierung zu erklären. Insgesamt denken junge Programmierende, dass Computer über die Fähigkeit verfügen, Befehle nicht nur zu verstehen, sondern auch den Sinn dahinter zu ergründen (Kohn & Komm, 2019). Pea (1986) beschrieb diese Annahme als «superbug» – Lernende nehmen an, Computer hätten eine intelligente Superkraft, die Code interpretieren kann.

2.3.2 Mathematik und Programmierkonzept «Variablen»

Oftmals bringen Lernende mathematische Vorstellungen in den Informatikunterricht ein (Sorva, 2018). In diesem Fachbereich begegnen sie ebenfalls dem Begriff «Variable» sowie Aufgabenstellungen, die mit diesem Thema einhergehen. Kohn und Komm (2019) beschreiben, dass Lernende Programmcode teilweise mit einer «mathematischen Brille» betrachten und Lösungsideen aus diesem Fach heranziehen. Aus mathematischer Sicht kommen Variablen oft in Gleichungen vor, die gelöst werden, indem eine unbekannte Zahl – eine Variable – herausgefunden wird. Dies kontrastiert mit der informatischen Definition des Begriffs, der zufolge eine Variable stets einen konkreten Wert besitzt (Kohn & Komm, 2019). FV 5, das ungelöste Ausdrücke gespeichert werden, hat ihren Ursprung in ebendiesem unterschiedlichen Verhalten von Variablen in den beiden Fachbereichen hat. Auch ist es in dieser Hinsicht nicht hilfreich, dass in manchen Programmiersprachen dasselbe Zeichen «=» beim Zuweisen eines Wertes verwendet wird wie in Gleichungen (Sorva, 2018). Das erklärt FV 8 und 9: Die Lernenden interpretieren die Zuweisung wie eine Gleichung in der Mathematik. In Programmcode gilt jedoch jeweils nur eine Zuweisungsrichtung von rechts nach links. Auch das Verständnis, dass Variablen ihren Wert abspeichern, kann nicht aus einer mathematischen Logik erklärt werden und ist somit für Lernende anspruchsvoll (FV 4).

Der Begriff «Variable» wird auch in anderen Fachbereichen verwendet (Kapitel 2.1.5). Diese unterschiedlichen Definitionen sind insbesondere für junge Lernende verwirrend und erschweren den Lernprozess (Luo et al., 2022).

2.3.3 Analogien für Variablen

Zur Veranschaulichung des Konzepts «Variablen» können Analogien verwendet werden (Kohn & Komm, 2019). Diese Analogien können jedoch auch Quellen von FV bei Lernenden sein (Hermans et al., 2018; Kohn, 2017; Sorva, 2018). In diesem Kapitel werden potenzielle Analogien für Variablen kurz beschrieben und teilweise illustriert. Gleichzeitig werden mögliche Bezüge zu FV hergestellt.

Box-Analogie (Hermans et al., 2018), auch: **Behältermodell** (Stechert, 2020b)

Bei dieser Analogie wird eine Variable als Box beschrieben. Der Name der Variable steht auf der Box und der Wert in der Box, beispielsweise auf einem Post-it (Hermans, o. J.). Der Wert der Variable kann in der Box auch mit Murmeln dargestellt werden (Hromkovič & Kohn, 2018). Stecher (2020b) stellt in seinem Video verschiedene Schwierigkeiten dieser Analogie vor:

- Wenn die Variable kopiert wird, ist die ursprüngliche Box anschliessend leer. (FV 6)
- Variablen müssen nicht initialisiert werden, weil die Box bereits vorhanden ist.
- Die Box speichert alle vorherigen Werte. (FV 1)
- Es spielt keine Rolle, welcher Datentyp in der Variable gespeichert wird.

Hermans et al. (2018) verglichen in einer Studie die Wirkung der Box-Analogie mit der Label-Analogie (vgl. weiter unten). Dabei stellten sie fest, dass bei Kindern und Erwachsenen, bei denen die Box-Analogie als Erklärungsstrategie verwendet wurde, das allgemeine Verständnis von Variablen gestärkt wurde, FV 1 jedoch vermehrt auftritt. Das Forschungsteam kam zu dem Schluss, dass die Box-Analogie zu Beginn des Lernprozesses hilfreich sein kann, Lernende jedoch in komplexeren Situationen verwirrt.

Label-Analogie (Hermans et al., 2018), auch: **Referenzierungsmodell** (Stechert, 2020b)

Bei dieser Analogie wird eine Variable als Label oder auch Etikett beschrieben, dem ein Wert zugeordnet wird. Diese Analogie kann beispielsweise mithilfe von Wäscheklammern bildlich umgesetzt werden, wie Abbildung 4 zeigt (Hermans, 2024). In der Studie von Hermans et al. (2018) zeigten die Teilnehmenden, die Variablen mit der Label-Analogie erlernten, schlechtere Resultate bei einfacheren Fragen, neigten jedoch in der Tendenz weniger zu FV 1.

Abbildung 4

Variable als Wäscheklammer gemäss



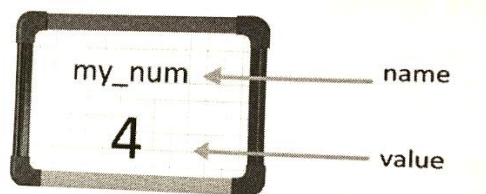
Anmerkung. Abbildung aus Hermans (2024)

Whiteboard-Analogie (Bagge & Grover, 2020)

Eine Variable kann auch mithilfe eines Whiteboards dargestellt werden (Abbildung 5). Dabei sind Wert sowie Name der Variable erkennbar und abwischbar. Dieses Merkmal unterscheidet diese Analogie von der Label-Analogie.

Abbildung 5

Variable als Whiteboard



Anmerkung. Abbildung aus Bagge & Grover (2020)

Weitere Analogien

In der Literatur sind auch Analogien zu finden, in denen ein bestimmter Aspekt von Variablen fokussiert wird. Beispielsweise erklärt Waguespack (1989), dass Variablen als Container mit unterschiedlicher Form nur Werte mit dem formlich entsprechenden Datentyp speichern können. Dieser Aspekt von Variablen kann auch mit der Parkplatz-Analogie erklärt werden (Pearce, 2024). Unterschiedliche Parkplatzarten (Datentyp der Variable) eignen sich dabei nur für bestimmte Fahrzeuge (Wert der Variable). Die Analogie «Eine Variable ist wie ein Fach in einem Regal»¹¹ eignet sich besonders bei der Verwendung der Programmiersprache BASIC (Hauswirth, 2024). Der Verlauf des Werts einer Variable kann auch mithilfe einer Tabelle¹² dargestellt werden. Jeder neu zugewiesene Wert wird dabei in der Tabelle unterhalb der zugehörigen Variable aufgeschrieben (Miller, 2024). Diese Art von Erklärung ist jedoch keine Analogie, sondern eine Repräsentation (Kapitel 2.2.6).

2.3.4 Rollen von Variablen

Um das Konzept der Variable besser zu verstehen, haben Forschende die Verwendung von Variablen in Programmen genauer betrachtet. Dabei hat sich herausgestellt, dass diese nicht zufällig verwendet werden, sondern innerhalb von wiederkehrenden Mustern auftauchen

¹¹ Engl. Pigeonhole

¹² Engl. Trace Table

(Kuittinen & Sajaniemi, 2004). Solche Muster werden als Rollen von Variablen bezeichnet. Diese Rollen beschreiben eine stereotypische Verwendung von Variablen in Computerprogrammen. Sajaniemi (2002) untersuchte Programme aus Lehrbüchern für Programmiernovizen und stellte fest, dass 99 % der in diesen vorkommenden Variablen einer von elf verschiedenen Rollen zugeordnet werden konnten. Diese Sammlung von Rollen wurde in verschiedenen Programmierparadigmen auf ihre Beständigkeit hin geprüft (Sajaniemi et al., 2006). Insgesamt konnten elf verschiedene Rollen verifiziert werden. In Tabelle 7 werden acht dieser Rollen beschreiben und anhand eines Beispiels illustriert. Auf die Beschreibung der Rolle des Organizers, Container, Walker wird an dieser Stelle verzichtet, da sich diese auf Datenstrukturen beziehen.¹³

Tabelle 7

Beschreibung acht möglicher Rollen von Variablen mit Beispiel nach

Rolle	Beschreibung	Beispiel
Fixed Value	Wert, der sich nach der Initialisierung nicht mehr verändert. Die Zuweisung kann im Programm oder über Benutzereingabe stattfinden.	Benutzername in einem Spiel
Stepper	Eine Variable, die eine Folge von Werten durchläuft, die vorhergesagt werden können, sobald die Folge beginnt. Die Variable kann beim Durchlauf des Programms größer oder kleiner werden. Der Abstand zwischen aufeinanderfolgenden Werten ist meist konstant.	Timer in einem Spiel
Most-recent-holder	Eine Variable, die den aktuellen Wert enthält, der beim Durchlaufen einer Folge von Werten gefunden wurde.	Antworten auf Quizfragen speichern
One-way flag	Eine Variable, die nur zwei Werte annehmen kann (bspw. true oder false).	Korrektur von Aufgaben anzeigen (richtig/falsch)
Most-wanted-holder	Eine Variable, die den bisher besten Wert enthält, der beim Durchlaufen einer Folge von Werten gefunden wurde. Der aktuelle Wert ist besser als alle vorherigen Werte.	Rekord in einem Spiel <ul style="list-style-type: none"> - weitester Wurf - schnellste Runde
Gatherer	Eine Variable, die einzelne Werte beim Durchlaufen einer Folge akkumuliert. Der neue Wert ergibt sich aus der Kombination mit dem vorherigen Wert.	Punktestand

¹³ Es gibt auch Arbeiten, welche noch weitere Rollen wie «Modifier» oder «Selector» definiert haben (Sajaniemi et al., 2006). Diese wurden jedoch hier nicht aufgenommen, weil sie zu komplex sind.

Follower	Eine Variable, die Werte einnimmt, die von einer anderen Variable abhängen, und meist direkt nach der Aktualisierung für die Aktualisierung einer anderen Variable verwendet wird.	Benutzereingaben
Temporary	Eine Variable, die den Wert nur für eine sehr kurze Zeit enthält. Der Wert ist eng verbunden mit anderen Variablen oder der Benutzereingabe.	Diese Variable wird gebraucht, wenn zwei Variablen ihre Werte tauschen.

Anmerkung. Eigene Darstellung in Anlehnung an Sajaniemi & Kuittinen (2006)

In einer Analyse von Lehrmitteln für die prozedurale Programmiersprache konnte Sajaniemi (2002) zeigen, dass die Rollen «Fixed Value», «Stepper» und «Most-recent-holder» 84 % aller untersuchten Variablen abdecken. In objektorientierten und funktionalen Programmiersprachen konnten «Fixed Value» und «Most-recent-holder» ebenfalls häufig identifiziert werden (Sajaniemi et al., 2006).

Sajaniemi und Kuittinen (2005) haben in einer Untersuchung die Wirkung der Erklärung dieser Rollen von Variablen in einem Kurs zur Programmiersprache Pascal angeschaut. Sie fanden heraus, dass erstens Studierende in der Lage sind, das Konzept der Rollen von Variablen zu verstehen und zweites diese Rollen es ihnen ermöglicht haben, Programme besser zu verstehen und zu verarbeiten.

2.3.5 Kompetenzaufbau für Variablen

Darüber, wann und wie Variablen in der obligatorischen Schulzeit unterrichtet werden sollen, besteht bislang Uneinigkeit. Auch existieren kaum empirische Hinweise darauf, wann aus entwicklungspsychologischen Gründen ein angemessener Zeitpunkt für die Einführung von Variablen ist.

Rich et al. (2022) veröffentlichten einen Lernpfad¹⁴, in dem der idealisierte und hypothetische Lernprozess zum Programmierkonzept «Variablen» basierend auf einer umfassenden Literaturanalyse zusammengefasst ist. Der Lernpfad beginnt auf der Stufe des *Data User*, auf der Lernende sich damit beschäftigen, was Daten sind und wie diese in einem Programm verwendet werden. Die nächste Stufe, *Data Storer*, beschreibt, dass Lernende verstehen, dass Variablen als Speicherplatz für Informationen dienen. Lernende verstehen auch den Code, der die Veränderung von Daten steuert, können diesen jedoch noch nicht selbst erstellen. Auf den ersten beiden Stufen nehmen die Lernenden die Sicht des Computers im Umgang mit

¹⁴ Engl. Learning Trajectory

Variablen und Daten ein. Auf der Stufe des *Variable User* wechseln die Lernenden zur Perspektive einer programmierenden Person. Sie beginnen, Variablen nun als Werkzeuge zu betrachten, um beispielsweise Benutzerangaben zu speichern. Sie können den Code verändern, der die Variable beeinflusst. Das Ziel besteht darin, die Stufe *Variable Creator* zu erreichen. Auf dieser Stufe können Lernende Variablen initialisieren und aktualisieren sowie verstehen, welchen Einfluss die Variable hat (Rich et al., 2020).

Luo et al. (2022) untersuchten diese Aktivitäten und die Entwicklung von Lernenden entlang des beschriebenen Lernpfads in einer Mixed-Methods-Studie. Dabei stellten sie fest, dass die meisten Lernenden nicht über die Stufe des *Data Storer* hinaus kamen und mit dem Verständnis von der Initialisierung und dem Updaten von Variablen sowie der Verwendung von Benutzereingaben Schwierigkeiten zeigten. Die Studie zeigt, dass das Verständnis des Programmierkonzepts «Variablen» mittels einer verstärkten Auseinandersetzung mit den spezifischen Eigenschaften von Variablen (vgl. Kapitel 2.1.4) gefördert werden kann. Buffum et al. (2018) fordern die Entwicklung von Aktivitäten zum Programmierkonzept Variablen, die im Unterricht isoliert genutzt werden können.

Es existieren noch keine empirischen Befunde, die den Kompetenzzuwachs entlang der beschriebenen Stufen beweisen oder widerlegen.

2.3.6 Diverse Befunde

In diesem Kapitel werden weitere Befunde und didaktische Hinweise zum Programmierkonzept «Variablen» in gesammelter Form dargestellt.

- Grover et al. (2019) entwickelten in einer Mixed-Methods-Studie interaktive Unplugged-Aktivitäten zum Programmierkonzept «Variablen» und stellten fest, dass diese Aktivitäten die korrekte Verwendung von Variablen in Programmierprojekten positiv beeinflussen.
- Van der Werf et al. (2023) untersuchten die Vermittlung von Variablen in Onlinekursen und stellten fest, dass die Box-Analogie oft Anwendung findet und damit die Speicherung von Variablen häufig thematisiert wird. Weiter beschreiben sie, dass der Benennung von Variablen besondere Beachtung geschenkt werden sollte, da diese für Lernende anspruchsvoll ist und bedeutungsvolle Namen von Variablen beim Verständnis von Code hilfreich sind.
- Lernende zeigten einen fähigeren Umgang mit Variablen vom Datentyp «String» als mit solchen vom Datentyp «Integer-Variablen». Dies könnte darauf zurückzuführen sein, dass im Umgang mit Zahlen in der Mathematik mehr Erfahrungen gesammelt wurden (Hermans et al., 2018). Es wird vorgeschlagen, dass altersentsprechende Namen für die verschiedenen Datentypen verwendet werden sollten. Vorgeschlagen wird Zahl für den Datentyp «Integer» und Text für den Datentyp «Strings» (Luo et al., 2022). Auch kann in Betracht

gezogen werden, zu Beginn des Lernprozesses nur einen Datentyp zu fokussieren (Bagge & Grover, 2020).

- In der systematischen Analyse von Zhang und Nouri (2019) über das Erlernen von CT-Kompetenzen mithilfe von Scratch sammelten die Forschenden Informationen zur Verwendung von Variablen in Scratchprojekten. Variablen werden in diesen genutzt, um einen Punktestand einzufügen (Vaníček, 2015), die Position, Entfernung oder Richtung einer Figur zu verändern (Falloon, 2016) oder geometrische Muster zu zeichnen (Plaza et al., 2018). Auch Paparo et al. (2021) haben in der Analyse von Scratchprojekten von Lernenden festgestellt, dass Variablen häufig als Punktestand oder Stoppuhr verwendet werden. In dieser Studie wurden den Lernenden vorgängig Programmbeispiele gezeigt, in denen unter anderen diese beiden Variablen vorkamen.

2.3.7 Zusammenfassung

Lernende haben häufig Schwierigkeiten, Variablen vollumfänglich zu verstehen und anzuwenden – unabhängig davon, ob sie mit block- oder textbasierten Programmiersprachen arbeiten. FV zu Variablen, wie die Annahme, dass eine Variable mehrere Werte gleichzeitig speichern könnte, erschweren das Lernen und resultieren aus der Anwendung mathematischer Gesetzmässigkeiten oder fehlerhafter Analogien (Kapitel 2.3.1). Mathematische Vorstellungen von Variablen widersprechen oft den informatischen Definitionen, was zu Missverständnissen führt (Kapitel 2.3.2). Analogien wie die Box- und Label-Analogie können das Verständnis erleichtern, jedoch auch neue FV hervorrufen (Kapitel 2.3.3). Forschungen zeigen, dass Variablen in Programmen spezifische Rollen einnehmen und ein Verständnis über diese Rollen den Studierenden das Lesen von Programmen erleichtern kann (Kapitel 2.3.4). Der Kompetenzaufbau für Variablen sollte strukturiert erfolgen, wobei es noch an empirischen Belegen für den idealen Zeitpunkt und die geeignete Methode mangelt (Kapitel 2.3.5). Weitere Studien heben die Bedeutung von Unplugged-Aktivitäten, der korrekten Benennung von Variablen und die Verwendung der Variable Punktestand in Spielen hervor (Kapitel 2.3.6).

3. Design und Methodik der Untersuchungen

3.1 Forschungslücke

Der aktuelle Forschungsstand zeigt auf, dass Variablen für Lernende anspruchsvoll und gleichzeitig für den Aufbau von Programmierkompetenzen relevant sind (Grover et al., 2019; Hermans et al., 2018; Kohn, 2017). Variablen werden von Lernenden selten in eigenen Projekten verwendet (Meerbaum-Salant et al., 2013). Analogien können dabei helfen, das abstrakte Konzept in einem den Lernenden bekannten Zusammenhang zu erklären (Fincher et al., 2020; Stechert, 2020b). Um darüber zu entscheiden, inwiefern eine Analogie für das Zielkonzept hilfreich ist, muss diese genau analysiert werden (Aubusson et al., 2006). Es sind unterschiedliche Analogien für das Programmierkonzept «Variablen» bekannt (Kapitel 2.3.3), wobei einige von ihnen bei Lernenden zu FV führen können. Es wurden jedoch noch nicht alle Analogien auf ihre Wirkung hin erforscht. Es gibt eine Sammlung von möglichen FV, die sich auf das Programmierkonzept «Variablen» beziehen (Kapitel 2.3.1), jedoch wurde bislang kaum untersucht, welche Vorstellungen – intuitiv oder elaboriert (Babari et al., 2023) – junge Lernende über Variablen haben. Ebendiese Perspektive der Lernenden sollte jedoch gemäss der didaktischen Reduktion (Diethelm et al., 2011) bei der systematischen Entwicklung von Unterricht miteinbezogen werden. Für das Programmierkonzept «Variable» wird die Entwicklung von Aktivitäten gefordert, bei denen sich verstärkt mit spezifischen Eigenschaften von Variablen auseinandergesetzt wird (Buffum et al., 2018; Luo et al., 2022). Im Folgenden wird der Informatikunterricht in Bezug auf das Programmierkonzept «Variablen» in den Blick genommen, um Hinweise darauf zu erlangen, wie dieses Konzept jungen Lernenden erklärt werden kann. Um dies zu erforschen, wird in dieser Masterarbeit der DBR-Ansatz verwendet. Im folgenden Abschnitt wird dieser kurz eingeordnet und erklärt.

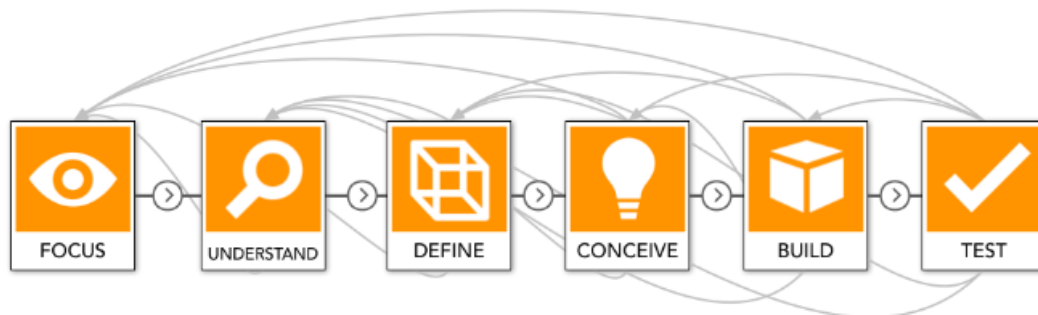
3.2 Design-Based-Research

DBR hat zum Ziel, den Transfer sowie die Umsetzung von Bildungsforschung in die Praxis zu erhöhen. Notwendig dafür sind die Bildung von Theorien sowie Gestaltungsprinzipien, die sowohl die Praxis als auch die Forschung leiten, informieren und verbessern sollen. DBR soll in der Bildungsforschung eine Brücke zwischen Praxis und Forschung schlagen (T. Anderson & Shattuck, 2012). Es wird ein wechselseitiges Verständnis gefordert, bei dem Forschung und Praxis nicht als dichotom verstanden werden, sondern der beeinflussende Charakter der beiden ins Zentrum gerückt wird (Euler, 2014). DBR ist somit als nutzungsorientierte Grundlagenforschung zu verstehen, in der weder Theorie noch Intervention für sich genommen ausreichen (Stokes, 1997).

Als zentrales Charaktermerkmal von DBR gilt die Iteration. Forschende sollen Resultate, Ideen oder Interventionen mit geeigneten Forschungsmethoden iterativ untersuchen und überarbeiten (Barab & Squire, 2004). Easterday, Lewis und Gerber (2014) haben unterschiedliche Definitionen dieses iterativen Prozesses miteinander verglichen und eine Definition für DBR formuliert, die die Grundlage dieser Arbeit bildet. Easterday et al. (2014) beschreiben DBR als einen Prozess, der aus sechs iterativen Phasen besteht, welche in Tabelle 8 beschrieben sind. Das bedeutet, dass der Ablauf nicht linear betrachtet werden kann und Erkenntnisse aus allen sechs Phasen eine Rückkehr zu einer vorherigen Phase bewirken können, was auch in Abbildung 6 ersichtlich ist.

Abbildung 6

Bildhafte Darstellung des iterativen DBR-Prozesses



Anmerkung. Darstellung aus Easterday et al. (2014)

Tabelle 8

Beschreibung der Phasen des DBR-Prozesses

Phase	Beschreibung
Focus	Thema, Zielpublikum und Umfang werden bestimmt.
Understand	Kernbereiche des Themas wie Lernende oder bestehende Lösungen werden untersucht und systematisiert zusammengetragen.
Define	Ziele für das Projekt werden festgelegt.
Conceive	Ein Entwurf/ein Plan für die Lösung des Problems wird entworfen.
Build	Die Lösung wird implementiert. Je nach Stand des Projekts variiert dabei die Genauigkeit.
Test	Die Wirkung der Lösung wird getestet.

Anmerkung. Eigene Darstellung in Anlehnung an Easterday et al. (2014)

Ziel von DBR ist aber nicht nur die Entwicklung einer Intervention, sondern auch das Definieren von Design-Prinzipien (DP). Diese sollen durch Reflexion des Forschungsprozesses entstehen und Hinweise darauf geben, was bei Interventionen in diesem Fachbereich relevant ist. DP gelten als heuristische Aussagen (Plomp, 2013).

3.3 Planung/Ablauf der Untersuchung

In diesem Kapitel wird die Planung beziehungsweise der Ablauf dieser Arbeit entlang der sechs Phasen beschrieben. Die Iterativität des Prozesses führte dazu, dass bei dieser Masterarbeit zu Beginn noch nicht ersichtlich war, worauf der Forschungsprozess abzielt. Der Forschungsprozess wurde an Ergebnisse aus den unterschiedlichen Phasen angepasst.

3.3.1 Focus

Im Rahmen der Masterarbeit wurde der Informatikunterricht in der Primarstufe untersucht. Genauer in den Fokus rückte dabei das Programmierkonzept «Variablen». Das Ziel bestand darin, herauszufinden, wie dieses Konzept Lernenden im Alter von 10–12 Jahren erklärt werden kann. Diese wurden als Zielgruppe definiert.

3.3.2 Understand

Um den Themenbereich «Variablen» und die didaktische Vermittlung desselben genauer zu verstehen, wurden Informationen auf drei verschiedenen Ebenen zusammengetragen. Die Grundlage bildete eine differenzierte Recherche zu aktuellen relevanten Forschungsbeiträgen zu fachdidaktischen Erkenntnissen im Informatikunterricht im Allgemeinen sowie zum Programmierkonzept «Variablen» im Besonderen. Zusammenfassungen der Ergebnisse sind im theoretischen Teil dieser Arbeit zu finden (Kapitel 2). Weiter wurde eine strukturierte Analyse bestehender Lehrmittel aus Deutschland, Österreich und Schweiz (DACH-Raum) sowie internationaler Forschungsbeiträge erstellt (Untersuchung 1, U1, Kapitel 4). Zusätzlich wurden Expertinnen und Experten aus der Fachdidaktik zu ihren Einschätzungen und Erfahrungen mit dem Programmierkonzept «Variablen» befragt (Untersuchung 2, U2, Kapitel 5).

Die Resultate aus U1 und U2 sowie der theoretische Hintergrund haben unter anderem gezeigt, dass Alltagsvariablen mögliche Analogien aus der Lebenswelt der Lernenden sein könnten, um das Programmierkonzept «Variablen» oder Eigenschaften desselben zu erklären. Daher wurde an dieser Stelle eine zweite Literaturrecherche durchgeführt, in der das Vorwissen und potenzielle Alltagsbezüge der Lernenden zum Thema «Variablen» im Fokus standen.

3.3.3 Define

Aus U1, U2 und der Literaturrecherche ergab sich das Ziel, herauszufinden, ob und ggf. welche Alltagsvariablen als mögliche Analogie für das Programmierkonzept «Variablen» im Unterricht mit Lernenden im Alter von 10–12 Jahren verwendet werden können. Der aktuelle Forschungsstand zur Vermittlung des Programmierkonzepts «Variablen» zeigte auch auf, dass Befunde zu möglichen FV zu finden sind (Kapitel 2.3.1). Bislang ist jedoch kaum erforscht, wie Lernende Variablen beschreiben und welche Vorstellungen sie über dieses Konzept haben. Als zweites Ziel wurde deshalb definiert, herauszufinden wie Lernende in diesem Alter das Programmierkonzept «Variablen» beschreiben.

3.3.4 Conceive

Um diese beiden Ziele beforschen, wurde ein Leitfadeninterview mit Lernenden der 6. Klasse durchgeführt. Dafür wurden Scratchprojekte entwickelt, in denen die Alltagsvariablen «Zeit», «Geschwindigkeit» und «Temperatur» sowie teilweise einer Variable für den Punktestand sichtbar waren. Die Lernenden wurden zuerst nach ihren Vorstellungen zu den Alltagskonzepten gefragt. Anschliessend durften sie die Scratchprojekte spielen und wurden schliesslich darum gebeten, zu beschreiben, was für sie «Variablen» sind. Die Alltagsanalogie und Scratchprojekte können als möglicher prototypischer Lösungsansatz zur Erklärung des Programmierkonzepts «Variablen» gesehen werden. Diese Prototypen wurden in den Interviews in einem ersten Durchgang untersucht. Die Ergebnisse werden in Kapitel 6 präsentiert.

Die Phasen «Build» und «Test» wurden im Rahmen dieser Masterarbeit aus zeitlichen Gründen nicht erreicht, können jedoch Ausgangspunkte zukünftiger Forschung sein.

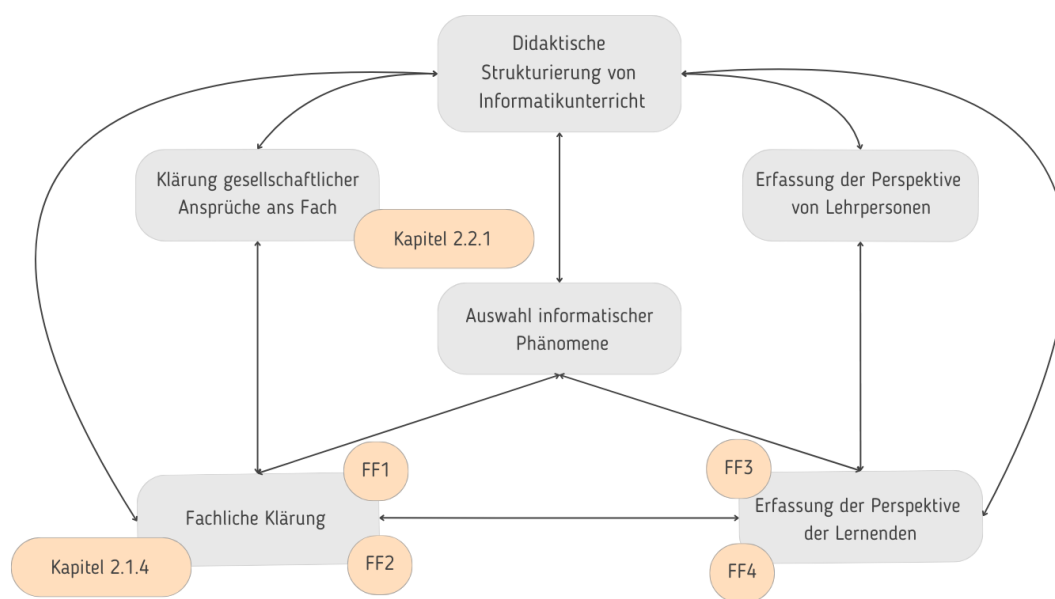
3.4 Didaktische Rekonstruktion «Variablen»

Neben dem methodischen Rahmen DBR ermöglicht die didaktische Rekonstruktion von Diethelm et al. (2011) eine didaktische Begründung der Untersuchungen dieser Masterarbeit. Aus Abbildung 7 geht hervor, dass die gesellschaftliche Relevanz im Rahmen des theoretischen Hintergrunds über die Bedeutung von CT aufgezeigt wurde. Für die fachliche Klärung sind einerseits die gesammelten Eigenschaften von Variablen in Tabelle 3 aus Kapitel 2.1.4 relevant. Andererseits ermöglichen U1 und U2 ein vertiefteres Verständnis darüber, welche Analogien, Aktivitäten und Rollen von Variablen im Unterricht eingesetzt werden können. Die Perspektive der Lernenden wird in U3 genau beleuchtet, um festzustellen, welche möglichen Alltagsbezüge zum Programmierkonzept «Variablen» hergestellt werden können und welche Vorstellungen Lernende zu diesem Begriff haben. Die Sammlung der FV (Kapitel 2.3.1) ist ebenfalls der Perspektive der Lernenden zuzuordnen. Es ist möglich, dass diese Arbeit erste Hinweise darauf gibt, welche informatischen Phänomene zu diesem Thema für junge Lernende

relevant sind. Die Perspektive von Lehrpersonen auf dieses Thema ist relevant, wird jedoch aus zeitlichen Gründen in dieser Arbeit nicht weiter thematisiert. Dies könnte jedoch einen Ausgangspunkt zukünftiger Forschung darstellen. Die Synthese der Ergebnisse dieser Arbeit liefert Erkenntnisse über potenzielle grundlegende Designprinzipien, auf deren Basis Informatikunterricht zum Thema «Variablen» für Lernende im Alter von 10–12 Jahren gestaltet werden sollte.

Abbildung 7

Einordnung der Masterarbeit in die Didaktische Rekonstruktion



Anmerkung. Eigene Darstellung in Anlehnung an Diethelm et al. (2011)

3.5 Verwendung von KI-Tools

In dieser Arbeit wurden verschiedene künstliche Intelligenzen für die im Folgenden aufgeführten Teilaufgaben verwendet.

- Für die Übersetzung der Einverständniserklärung, des Interviewleitfadens sowie der Forschungsartikel wurde DeepL¹⁵ verwendet.

¹⁵ DeepL Zugriff via: <https://www.deepl.com/de/translator>

- ChatGPT¹⁸ wurde genutzt, um Forschungsfragen zu schleifen, über Fachtexte zu diskutieren, Unterkategorien in Interviews zu überprüfen und Textteile stilistisch und formal zu kontrollieren.

¹⁸ ChatGPT Version 4.0 Zugriff via: <https://chatgpt.com/>

4. Untersuchung 1: Analyse Lehrmittel und Forschungsbeiträge

Im Prozess des DBR ist es elementar, den fokussierten Themenbereich genau zu untersuchen. Bereits bestehende Lösungsansätze sollen in der Phase «Understand» ermittelt und analysiert werden (Easterday et al., 2014). In diesem Kapitel wird eine Untersuchung (U1) der bestehenden Lehrmittel und Forschungsbeiträge vorgestellt. Dazu wird die Fragestellung (Kapitel 4.1) erläutert und die Stichprobe (Kapitel 4.2), die Untersuchungsmaterialien (Kapitel 4.3), die Prozedur (Kapitel 4.4) sowie der Ablauf der Datenauswertung (Kapitel 4.5) werden beschrieben. Die Ergebnisse aus U1 werden in Bezug zur zugrunde liegenden Forschungsfrage gesetzt (Kapitel 4.6) und anschliessend diskutiert (Kapitel 4.7).

4.1 Forschungsfrage und Erwartungen

Es existieren kaum Untersuchungen darüber, wie Variablen unterrichtet werden (Van Der Werf et al., 2023). Um Einblicke in die aktuelle Unterrichtspraxis zu erlangen und damit bestehende Ansätze zur Vermittlung des Programmierkonzepts «Variablen» zu analysieren, werden in U1 Lehrmittel und Forschungsbeträge untersucht. Die zugehörige FF1 lautet wie folgt:

FF1: Welche Analogien, Rollen und Aktivitäten zur Veranschaulichung des Programmierkonzepts «Variablen» finden in Informatiklehrmitteln im DACH-Raum und in internationalen Forschungsbeiträgen für Lernende der Klassen 5–7 Anwendung?

Zur FF1 sind in Bezug auf die theoretische Grundlage dieser Arbeit Erwartungen formuliert worden. Einerseits wird erwartet, dass die in Kapitel 2.3.3 vorgestellten Analogien in den Lehrmitteln und Forschungsbeiträgen vorkommen und dabei die Box-Analogie am häufigsten genutzt wird, wie Hermans et al. (2018) belegen. Gemäss Sajaniemi (2002) decken die Rollen «Fixed Value», «Stepper» und «Most-recent-holder» 84 % der Variablen ab, die in Aufgaben aus Lehrmitteln vorgekommen sind. Obwohl sich in dieser Studie mit prozeduralen Programmen befasst wurde, wird erwartet, dass diese drei Rollen in U1 ebenfalls häufig vorkommen. Hinsichtlich der verwendeten Aktivitäten fehlt eine theoretische Grundlage zur Formulierung einer Erwartung. Auf der Basis der Unterrichtserfahrung der Autorin auf der Zielstufe von U1 lässt sich jedoch annehmen, dass weniger Aktivitäten vorhanden sind, die einen Unplugged-Zugang zu Variablen ermöglichen. Die Autorin erlebt es häufig, dass Lehrmittel zahlreiche Aufgaben mit einer Umsetzung in einer Programmierumgebung bereitstellen, jedoch selten Ideen für Unplugged-Übungen aufzeigen.

Um FF1 zu beantworten, wird eine inhaltliche Datenanalyse durchgeführt, bei der der Fokus auf die Erfassung von Sachordnungen innerhalb der Lehrmittel und Forschungsbeiträge im Bereich des Programmierkonzepts «Variablen» gelegt wird (Hoffmann, 2018, S. 21). Dabei wird in Anlehnung an Denz (1989) vorgegangen: Nach der Aufstellung einer theoriegeleiteten Forschungsfrage wird eine repräsentative Auswahl des Materials erstellt. Danach folgen die Festlegung der Kategorien und möglichen Merkmalsausprägungen. Teilweise ist eine feinere Bestimmung geeigneter Untersuchungseinheiten innerhalb des Materials erforderlich. Es folgt eine Zuordnung der Untersuchungseinheiten zu den definierten Kategorien. Die Ergebnisse werden anschliessend deskriptiv in Bezug auf die Forschungsfrage ausgewertet. Zuletzt werden die Resultate interpretiert und FF1 beantwortet.

4.2 Stichprobe

Um FF1 zu beantworten, werden Forschungsbeiträge sowie in Lehrmitteln publizierte Unterrichtsreihen untersucht.

Lehrmittel

Bei der Auswahl der Lehrmittel wurden solche aus der Schweiz, Deutschland und Österreich berücksichtigt. Aus organisatorischen und finanziellen Gründen wurden für Deutschland und Österreich nur Lehrmittel berücksichtigt, die über einen Onlinezugang abgerufen werden konnten. Für die Schweiz wurden sowohl die offiziellen Lehrmittel (Kommission für Lehrplan- und Lehrmittelfragen, 2023) als auch Veröffentlichungen von pädagogischen Hochschulen berücksichtigt. Wenn vorhanden, wurde auch dazugehöriges Zusatzmaterial, wie Handreichungen und Lehrerkommentare, berücksichtigt.

Forschungsliteratur

Bei der systematischen Recherche zu Forschungsliteratur zum Programmierkonzept «Variablen» diente die Studie von Rich et al. (2020) als Ausgangspunkt, da das Forschungsteam relevante Literatur zu dieser Thematik zusammengestellt hat. Ergänzend wurde auf den Datenbanken ERIC, Researchgate, ScienceDirekt und Google Scholar mit den Suchbegriffen «teaching variables in computer science», «analogies for variables in computer science education» sowie abgewandelten und deutschen Formen Literatur ermittelt. Forschungsbeiträge wurden in U1 integriert, wenn sie einen Fokus auf die Vermittlung des Programmierkonzepts «Variablen» auf der Zielstufe enthielten. Wenn vorhanden, wurde dazugehöriges Unterrichtsmaterial ebenfalls in die Stichprobe aufgenommen. In diesem Fall wurde der Beitrag als Mischform kategorisiert.

In Tabelle 9 sind die zehn Lehrmittel, vier Forschungsbeträge und drei Mischformen aufgelistet, die für U1 ausgewählt wurden. Eine Übersicht inklusive Zugriffsmöglichkeit ist Anhang

10.1.1 zu entnehmen. Fünf Lehrmittel kommen aus der Schweiz, fünf aus Deutschland (wobei diese teilweise auch in Österreich angewendet werden) und eines wurde in Österreich entwickelt. Von den Forschungsbeiträgen und Mischformen stammen fünf aus Amerika; jeweils ein Beitrag stammt aus den Niederlanden und aus Deutschland.

Tabelle 9

Übersicht Stichprobe U1 (n = 17)

Nr.	Titel	Verlag / Herausgeber	Land	Art
L1	Connected 3	LMVZ	CH	L
L2	Einfach Informatik 7–9	Klett und Balmer Verlag	CH	L
L3	Gruppenpuzzle	PHSZ	CH	L
L4	PaperPenny	PHBern	CH	L
L5	Vernetzt – digitale Grundbildung	öbv	Ö	L
L6	Imperative Programmiersprache – Algorithmisches Problemlösen mit Kara	inf-schule	D	L
L7	Imperative Programmiersprache – Algorithmisches Problemlösen mit Scratch	inf-schule	D	L
L8	Einfach Programmieren	Carlsen	D	L
L9	Duden Informatik SEKI Baden -Würt-tenberg	Cornelsen	D	L
L10	Informatik 56 Nordrhein-Westfalen	Cornelsen	D	L
F11	Variable Evaluation: an Exploration of Novice Programmers' Understanding and Common Misconceptions	(Kohn, 2017)	D	F
M12	Act 2 Curriculum	ECforAll	A	M
M13	Robot Boxes	Everyday Computing	A	M
F14	Variables	(Bagge & Grover, 2020)	A	F
M15	Vela	(Grover et al., 2019)	A	M

F16	Introducing the Computer Science Concept of Variables in Middle School Science Classrooms	(Buffum et al., 2018)	A	F
F17	Thinking out of the box	(Hermans et al., 2018)	N	F

Anmerkungen: CH steht für Schweiz, Ö für Österreich, D für Deutschland, A für Amerika und N für Niederlande. L steht für Lehrmittel, F steht für Forschungsbeitrag und M für Mischform.

4.3 Untersuchungsmaterialien

Für die qualitative Analyse wurde ein Kriterienraster entwickelt, um Lehrmittel und Forschungsbeiträge systematisch zu untersuchen und zu vergleichen. Die Kriterien wurden deduktiv, ableitend aus FF1, gebildet. Die Kriterien der Rollen von Variablen wurden aus der Theorie von Sajaniemi et al. (2006) bestimmt und Themenbereiche des Programmierens nach Zhang und Nuri (2019) abgeleitet. Die restlichen Ausprägungen haben sich induktiv aus den Erkenntnissen aus der Analyse ergeben. Das endgültige Kriterienraster kann Anhang 10.1.2 entnommen werden. Im Folgenden werden die in Bezug auf FF1 relevanten Kategorien genauer beschrieben.

4.3.1 Analogien für Variablen

In dieser Kategorie steht im Fokus, wie Variablen den Lernenden im Lehrmittel oder Forschungsbeitrag erklärt wurden. Dazu wurde in diesen nach verwendeten Analogien gesucht. Eine Analogie wurde hier kategorisiert, wenn sie beschrieben oder visualisiert wurde. Zusätzlich wurde teilweise ein Kommentar verfasst, wenn beispielsweise im Begleitband für Lehrpersonen Hintergründe zum Einsatz der Analogie verfasst wurden oder im Forschungsprojekt Ergebnisse zur Verwendung der Analogie beschrieben wurden.

4.3.2 Rollen von Variablen

Für diese Kategorie wurden die Ausprägungen entsprechend der acht von Kuittinen und Sajaniemi (2004) beschriebenen Rollen von Variablen übernommen. Eine genaue Beschreibung der Rollen inklusive Beispiele ist in Tabelle 7 zu finden. Diese Kategorie wurde nur bei Lehrmitteln und Mischformen berücksichtigt, da in den Forschungsbeiträgen meist keine Aufgaben oder Beispiele dargestellt wurden. Pro Unterrichtsbeispiel wurden alle vorkommenden Variablen einer Rolle zugewiesen, weshalb eine Unterrichtseinheit mehrere Rollen beinhalten kann. Es erfolgte jedoch keine Mehrfachzählung, wenn eine Rolle in mehr als einem Beispiel verwendet wurde. Je nach Aufgabenstellung oder Zusatzaufgabe ist es möglich, dass sich die Rolle einer Variable ändert oder eine neue dazu kommt. In diesem Fall wurden beide Rollen

erfasst. Der Prozess des Kategorisierens der Rollen von Variablen war herausfordernd, da es ein grundlegendes informatisches Verständnis erforderte. Als hilfreich wurde dabei das Erklärvideo von Stechert (2020a) erachtet. Die Beschreibungen in diesem Video flossen auch in die Formulierungen im Kategoriensystem ein.

4.3.3 Aktivitäten

«Aktivitäten» ist ein weitgefasster Begriff. Um diesen innerhalb des Kategoriensystems erfassbar zu machen, wurde erhoben, welchem Themenbereich nach Zhang und Nouri (2019) die Unterrichtsreihe zugeordnet werden konnte. Zu diesen Themenbereichen gehörten die Ausprägungen «Storytelling», «Musik», «Animation», «Mixed» und «Game». Diese Liste wurde um die Ausprägung «Mathematik» ergänzt, da diese in mehreren Lehrmitteln gefunden wurde. Zudem wurde erfasst, welche Programmierlernumgebung in dieser Unterrichtsreihe verwendet wurde. Abschliessend wurde geprüft, welchen Charakter die erwähnten Aktivitäten haben, also ob diese plugged oder unplugged gestaltet sind oder ob beide Arten vorkamen. Unplugged-Aktivitäten wurden stichwortartig beschrieben.

4.4 Prozedur

Die Suche nach passenden Lehrmitteln und Forschungsbeiträgen fand zwischen September 2023 und Januar 2024 statt. Die Lehrmittel wurden aus dem Didaktischen Zentrum Uri ausgeliehen, über einen digitalen Zugang gesichtet oder – in Ausnahmefällen – angeschafft. Für die Lehrmittel aus Deutschland und Österreich wurden Dozierende aus den jeweiligen Ländern per E-Mail um Auskunft über mögliche Lehrmittel gebeten. Die Forschungsbeiträge wurden wie in Kapitel 4.2 beschrieben gesucht und heruntergeladen. Unter Berücksichtigung des Kategoriensystems wurden die Lehrmittel und Forschungsbeiträge geprüft und analysiert.

4.5 Ablauf der Datenauswertung

Das Vorkommen der Ausprägungen aus dem Kategoriensystem (Anhang 10.1.2) wurde in einer Excel-Liste erfasst. Die vollständige Auswertung kann dem Anhang 10.1.3 entnommen werden. Zur Erstellung deskriptiver Ergebnisse wurden Auszählungen der Ausprägungen durchgeführt und anschliessend in Excel Diagramme erstellt.

4.6 Ergebnisse

In diesem Kapitel werden die Ergebnisse zu U1 präsentiert und in Bezug zu FF1 «Welche Analogien, Rollen und Aktivitäten zur Veranschaulichung des Konzepts «Variable» finden in Informatiklehrmitteln im DACH-Raum und in Forschungsbeiträgen für Lernende der Klassen 5–7 Anwendung?» gesetzt. Bei der Darstellung der Ergebnisse wird werden Abkürzungen

(Lehrmittel = L, Forschungsbeitrag = F, Mischform = M) und die entsprechenden Nummerierungen aus Tabelle 9 verwendet.

4.6.1 Verwendete Analogien für Variablen

In den Lehrmitteln und Forschungsbeiträgen wurden unterschiedliche Analogien zur Erklärung des Programmierkonzepts «Variablen» verwendet (Abbildung 8). Wenn eine Analogie verwendet wurde, kam die Box-Analogie am häufigsten vor. Potenzielle Veranschaulichungen dieser Analogie sind in Abbildung 9 und Abbildung 10 ersichtlich. Im digitalen Handbuch des L1 wird die Umsetzung der Box-Analogie mit einer realen Schachtel und Wertekarten vorgeschlagen. Zusätzlich werden die Lehrpersonen über mögliche FV informiert, die in Bezug auf die verwendete Analogie entstehen können. Im Begleitband zu L2 wird über Problematiken der Box-Analogie berichtet und vorgeschlagen, selbige nur mit Zahlvariablen zu verwenden und den Wert jeweils physisch, beispielsweise mit Murmeln, darzustellen. Im F17 wird die Wirkung der Box-Analogie mit jener der Label-Analogie verglichen.

Abbildung 8

Übersicht der verwendeten Analogien in ausgewählten Lehrmitteln und Forschungsbeiträgen (n = 17)

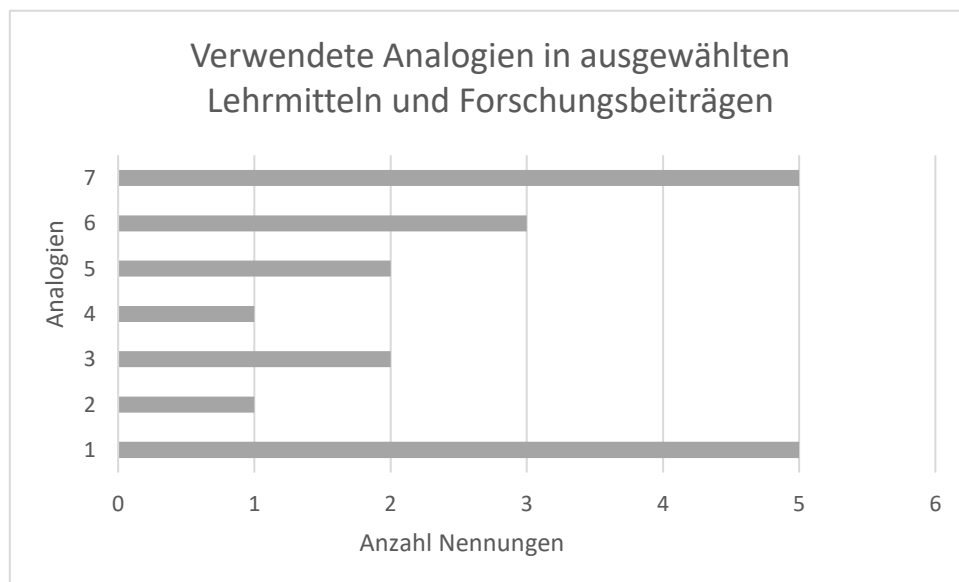


Abbildung 9

Veranschaulichung Box-Analogie in L1



Abbildung 10

Veranschaulichung Box-Analogie in L12, hier mit einem Glas



In drei Lehrmitteln wurden zur Veranschaulichung der Werte von Variablen Tabellen verwendet. Abbildung 11 zeigt ein Programm, das eine Summe berechnet. In diesem Programm verändern sich die Werte der Variable `«Summe»` und der Variable `«x»`. Diese Veränderungen können in einer Tabelle dargestellt werden, was in Abbildung 12 visualisiert ist. Im Begleitband des L2 ist beschrieben, dass in der Programmiersprache Python tatsächlich eine Art von Tabelle verwendet wird, um Variablen zu organisieren.

Abbildung 11

Programm mit den Variablen `«Summe»` und `«x»` aus L2

```
summe = 0
x = 1
repeat 100:
    summe += x
    x += 1
print summe
```

Abbildung 12

Tabelle zur Repräsentation der Variablen aus Programm in Abbildung 11 aus L2

Zeit-Tabelle							
i-ter Durchlauf der Schleife	1	2	3	4	5	6	...
summe	0	1	3	6	10	15	...
x	1	2	3	4	5	6	...

In zwei Lehrmitteln wurde das Programmierkonzept `«Variablen»` mithilfe einer oder mehrerer Alltagsanalogien erklärt. L9 verwendet dafür das Alter einer Person (Abbildung 9). Bei M15 werden unterschiedliche Beispiele aus dem Alltag von Lernenden verwendet, in denen jeweils die Variable visualisiert wird. Hierbei kommen als Alltagsvariablen die Temperatur, die T-Shirt-Größe, der Punktestand bei Sportereignissen und der Preis eines Gegenstands vor.

In L8 wird als Analogie zur Variable ein Briefumschlag herangezogen. Den Lernenden wird erklärt, dass der Name der Variable als Beschriftung auf dem Umschlag steht und der Wert der Variable auf einem Zettel in den Umschlag gehört.

Die Label-Analogie kam zweimal in den untersuchten Materialien vor. In M13 wird für die Visualisierung der Variable ein Kasten verwendet, auf dem ein Post-It angebracht werden kann.

Ein Beispiel einer solchen Variable ist in Abbildung 14 ersichtlich. In F17 wird die Wirkung der Label-Analogie mit jener der Box-Analogie verglichen.

In F14 wird erwähnt, dass zur Erklärung des Programmierkonzepts «Variablen» ein Whiteboard genutzt werden kann.

Abbildung 13

Verwendung der Alltagsvariable «Alter» in Lehrmittel 9



Abbildung 14

Veranschaulichung der Label-Analogie in Beitrag 13



In fünf Beiträgen konnte keine Analogie identifiziert werden. Variablen wurden in diesen Beiträgen mit einer Definition erklärt, wobei hier keine analoge Beziehung festgestellt werden konnte. In L5 wird zuerst das Konzept «Variablen» für die Mathematik erklärt und anschliessend Folgendes über Variablen festgehalten: «Im Computerbereich werden ebenfalls Variablen verwendet. Der Variablenname darf aus mehr als einem Buchstaben bestehen.» (Fikisz et al., 2023, S. 235)

4.6.2 Vorkommende Rollen von Variablen

In den Beispielen und Aufgaben der untersuchten 12¹⁹ Lehrmittel und Forschungsbeiträge sind drei unterschiedliche Rollen von Variablen gefunden worden: «Fixed Value», «Stepper» und «Gatherer». Die anderen Rollen von Variablen, wie «Most-recent-holder» wurden in den Beispielen und Aufgaben nicht gefunden²⁰. Im Folgenden werden die drei vorkommenden Rollen mit Beispielen aus den Lehrmitteln und Forschungsbeiträgen beispielhaft illustriert.

¹⁹ Bei 5 Beiträgen konnten die genauen Aufgaben nicht eingesehen werden, weshalb diese hier ausgeschlossen wurden.

²⁰ Da jeweils nur das einmalige Vorkommen der Rolle gewertet wurde und mehrfachvorkommen nicht in die Analyse einfließen wird an dieser Stelle darauf verzichtet eine genau Angabe über die Anzahl gefundener Rollen zu machen. Dies kann jedoch der Auswertung im Anhang 10.1.3 entnommen werden.

Eine Variable, die als «Fixed Value» verwendet wurde, ist die Variable «meinName» in L8. In L3 speicherte die Variable eine Zufallszahl, die sich im Verlauf des Programms nicht verändert. In F14 wurden die Länge und die Breite und in L5 zusätzlich die Höhe für die Berechnung von Fläche oder Volumen von Körpern verwendet. In L9 und L10 wurde die Schrittlänge für eine Figur gespeichert.

Eine Variable, die als «Gatherer» verwendet wurde, ist die Punktezah in einem Spiel, die in L1, L3, L9 und M12 Anwendung findet. In diesen Spielen konnte die Punktzahl meistens durch eine Aktion erhöht werden. In L5 wurde das Volumen eines Quaders berechnet; in M13 wurden die Fläche und der Umfang eines Rechtecks mithilfe der Grundoperationen Multiplikation und Addition kalkuliert. In L2 und L6 wurde die Variable «Summe» verwendet.

Die Variable «Zähler» in L6 und L7 wurde der Rolle des Steppers zugewiesen. Im L7 zählte dieser Zähler die Sekunden in einem Spiel. In L9 wurde die Variable «Alter» jährlich um eins erhöht. Im Beispiel von Abbildung 11 aus L2 funktionierte die Variable x als Stepper, der mit jedem Durchlauf einer Schleife um eins erhöht wird.

4.6.3 Aktivitäten für Variablen

Es wurde ermittelt, welcher thematische Zugang in den Lehrmitteln und Forschungsbeiträgen gewählt wurde. Am häufigsten wurden Variablen innerhalb eines Spiels thematisiert, was Abbildung 15 zeigt. In L3 wurde mit einem Spiel gearbeitet, bei dem eine Zufallszahl erraten werden muss. In L4 wurde mit einem Jump 'n' Run-Spiel gearbeitet. In M12 stand ein Fussballspiel im Fokus, bei dem Tore geschossen werden müssen.

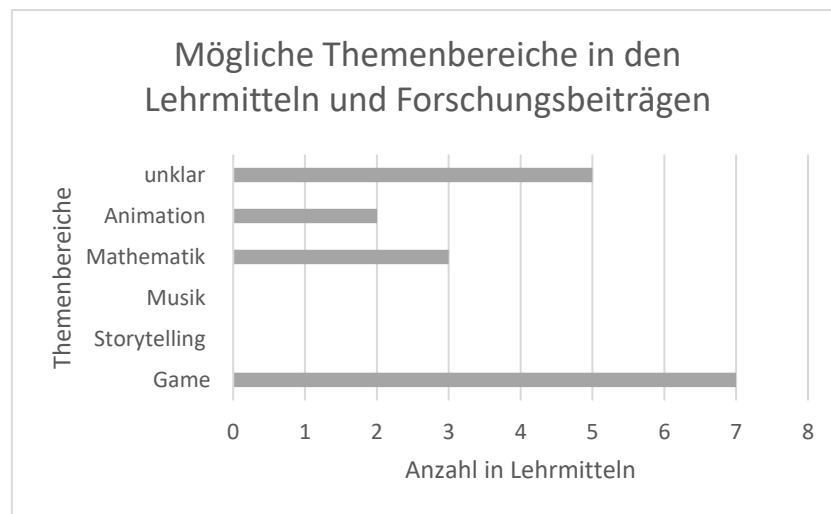
Dreimal wurde im untersuchten Beitrag ein Bezug zur Mathematik hergestellt. Wie bereits beschrieben, wurden in L5 und M13 geometrische Berechnungen mithilfe von Variablen gelöst. In L2 wurden Variablen unter anderem verwendet, um geometrische Figuren zu zeichnen.

In zwei Beiträgen wurden Programme vorgestellt, die eine Animation eines Ablaufs zeigten. L1 zeigte die Programmierung einer Drohne, wobei die Variable den Akku selbiger speichern soll. In F16 wurde eine Aktivität vorgestellt, bei der Lernende einen Rechner für die Grösse der Bevölkerung programmieren.

In fünf Beiträgen konnte kein klarer Themenbereich kategorisiert werden. Vier Beiträge davon stammen aus der Forschung. Die Themenbereiche «Musik» und «Storytelling» wurden in den untersuchten Beiträgen nicht verwendet.

Abbildung 15

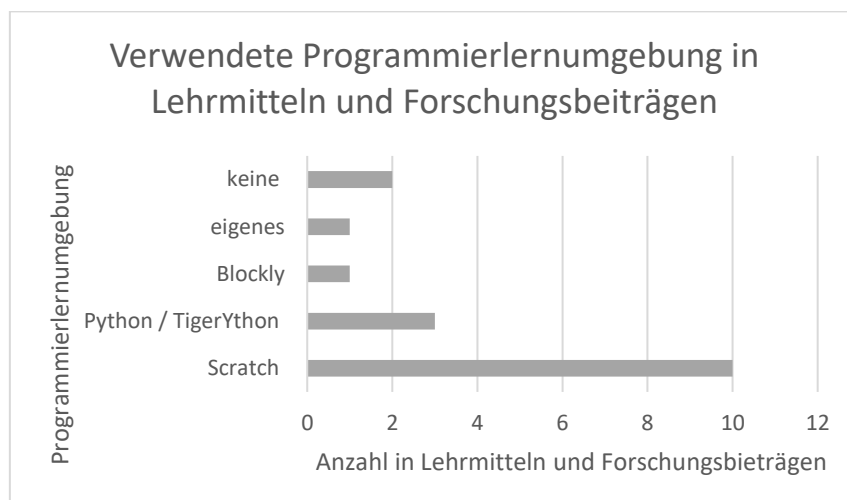
Einordnung der Aktivitäten, die in ausgewählten Lehrmitteln und Forschungsbeiträgen vorkamen, in Themenbereiche (n = 17)



Es wurde auch geprüft, welche Programmierlernumgebung die Beiträge verwendet haben, was in Abbildung 16 zu sehen ist. 10 Beiträge haben die Programmierlernumgebung «Scratch» verwendet. Dabei wurden einerseits bildhafte Beispiele mit Scratchblöcken gezeigt und/oder Aufgaben mit einer Umsetzung in Scratch gestellt. In drei Lehrmitteln wurde die Programmiersprache Python verwendet, wobei in L2 und F17 die Programmierlernumgebung TigerJython²¹ genutzt wurde. In F16 wurde die Programmierlernumgebung Blockly²² verwendet; in L8 wurde ein eigenes App integriert. In F14 und M15 wurde keine Programmierlernumgebung verwendet.

Abbildung 16

Übersicht über verwendete Programmierlernumgebungen in Lehrmitteln und Forschungsbeiträgen (n = 17)



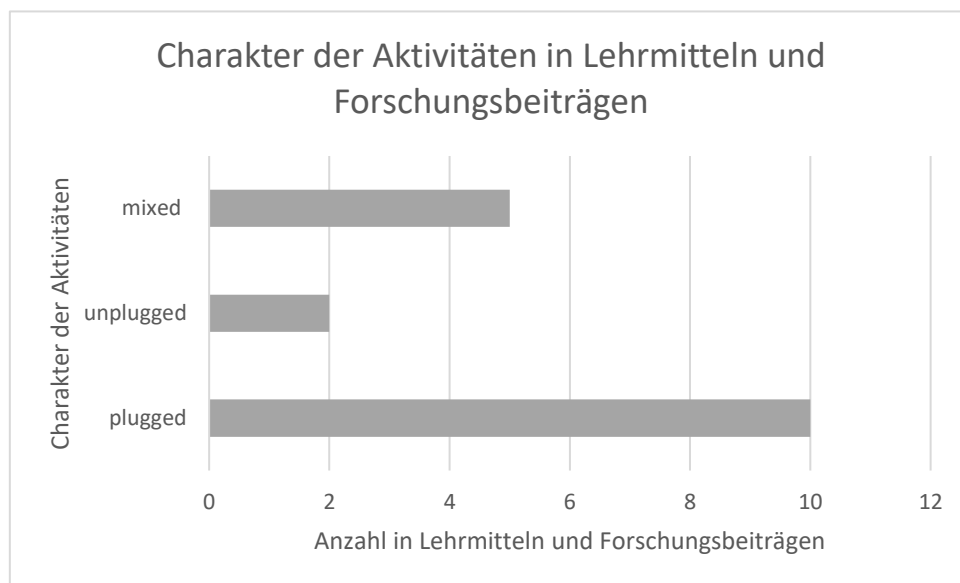
²¹ TigerJython erreichbar unter : <https://tigerjython.ch/de>

²² Blockly erreichbar unter: <https://developers.google.com/blockly?hl=de>

Zudem wurde ausgewertet, welchen Charakter die Aktivitäten der Beiträge hatten, was in Abbildung 17 zu sehen ist. In 10 Lehrmitteln und Beiträgen wurden nur auf Aktivitäten gefunden, die mit einem Gerät umgesetzt werden müssen. Dabei handelt es sich meistens um Programmieraufgaben. In M12 wurden Lernende aktiv dazu aufgefordert, bestehende Programme zu spielen sowie Code zu lesen und diesen zu verändern. In L1, L6, L8, M12 und M13 wurden sowohl Unplugged- als auch Plugged-Aktivitäten verwendet; in F14 und M15 wurde sich auf Unplugged-Aktivitäten beschränkt.

Abbildung 17

Übersicht über den Charakter der Aktivitäten in Lehrmitteln und Forschungsbeiträgen ($n = 17$)



Es wurden sieben Unplugged-Aktivitäten gefunden, die in Tabelle 10 zusammengefasst dargestellt werden.

Tabelle 10

Beschreibung der Unplugged-Aktivitäten in den untersuchten Lehrmitteln und Forschungsbeiträgen

Name	Beschreibung	Quelle
Schachtelübung	Die Variable im Programm soll mithilfe einer Schachtel und Wertekarten nachgespielt werden.	L1
Rollenspiel	Die Lernenden übernehmen die Rolle des/der Datenverwalter:in, der/die den Überblick über die Variablen hat und Prozessor, der das Programm verarbeitet.	L6

Adressübung	Die Lernenden erhalten drei Briefumschläge, die sie mit «Frühstück», «Mittagessen» und «Abendessen» beschriften. Auf kleineren Zetteln wird notiert, was bei der jeweiligen Mahlzeit gegessen wurde. Diese drei Variablen sollen jeden Tag aktualisiert werden.	L8
Kartenspiel VARIO ²³	Das Spiel hat einen ähnlichen Grundaufbau wie Uno mit Zahlenkarten in vier verschiedenen Farben und Spezialkarten. Spieler:innen halten den eigenen Punktestand in einer Tabelle fest. Der Punktestand kann sich vergrößern durch das Legen einer Zahlenkarte oder verkleinern durch das Legen einer Spezialkarte.	M12
Robot Boxes	Die Lernenden berechnen die Fläche und den Umfang eines Rechtecks mithilfe der Variablen «Länge» und «Breite». Dabei übernehmen die Lernenden die Rolle des Roboters, der das Programm abspielt, des Fillers, der das Protokollblatt ausfüllt, und des Calculators, der die Berechnungen löst. Die Variablen werden nach jedem Durchgang verändert.	M13
Whiteboardübung	Die Variable soll in einem einfachen Programm mithilfe eines Whiteboards dargestellt werden. Lernende können selbst eigene Beispiele erstellen, die andere Lernende nachspielen.	F14
Story Variables	Lernende erhalten Bilder von alltäglichen Situationen und vergleichen diese miteinander, suchen Variablen und überlegen sich passende Werte für diese Variablen.	F14, M15

Bei der Schachtel-, Adress- und Whiteboardübung wurde die verwendete Analogie in einer realen Aufgabenstellung fokussiert. Bei der Aktivität «Robot Boxes» wurden auf dem Protokollblatt Post-its für die Werte der Variablen verwendet. Diese Aktivität wurde als Visualisierung der Label-Analogie kategorisiert. Beim Rollenspiel und beim Kartenspiel VARIO wurde die Repräsentationsform der Tabelle verwendet, um die Veränderung des Werts einer Variable darzustellen.

4.7 Zwischendiskussion und Fazit

In diesem Kapitel wird die FF1 beantwortet und darauf eingegangen, wie die erzielten Ergebnisse im Kontext der vorliegenden Arbeit zu deuten sind.

²³ Videoanleitung für das Spiel Vario: <https://bit.ly/VARIOgame>

4.7.1 Beantwortung von FF1

Um FF1 «Welche Analogien, Rollen und Aktivitäten zur Veranschaulichung des Programmierkonzepts «Variablen» finden in Informatiklehrmitteln im DACH-Raum und in internationalen Forschungsbeiträgen für Lernende der Klassen 5–7 Anwendung?» zu beantworten, wurden Lehrmittel und Forschungsbeiträge mittels Kategoriensystem analysiert. Anhand der Ergebnisse dieser Analyse kann FF1 mit drei Teilantworten beantwortet werden.

- In den untersuchten Lehrmitteln aus dem DACH-Raum und in internationalen Forschungsbeiträgen werden die Box-Analogie, die Alltagsanalogie, die Label-Analogie, die Briefumschlags-Analogie und die Whiteboard-Analogie angewendet. Die Repräsentationsform der Tabelle wird auch genutzt, gehört jedoch nicht in die Kategorie «Analogie» gemäss Miller (2024). Die Erwartung, dass die Box-Analogie in U1 am häufigsten vorkommt, wurde bestätigt, was mit den Aussagen von Hermans et al. (2018) übereinstimmt. Die Erwartung, dass alle in Kapitel 2.3.3 vorgestellten Analogien vorkommen, hat sich nur teilweise bestätigt. Die Alltagsanalogie und die Briefumschlags-Analogie tauchten in dieser U1 zusätzlich auf.
- In den untersuchten Lehrmitteln aus dem DACH-Raum und in den internationalen Forschungsbeiträgen kamen Variablen in den Rollen «fixed Value», «Stepper» und «Gatherer» vor. Dies entspricht nur teilweise den in den Erwartungen thematisierten Rollen. Im Kontrast zu den Resultaten von Sajaniemi (2002) kommt in U1 die Rolle des «Gatherers» häufig vor; die Rolle «Most-recent-holder» wurde in keiner Aktivität lokalisiert. Die Abweichung könnte damit zu tun haben, dass die von Sajaniemi untersuchten Programme in Lehrbüchern für ältere Studierende vorkamen.
- Die meisten Aktivitäten, die bei der Einführung des Themas «Variablen» in den untersuchten Lehrmitteln aus dem DACH-Raum und in den internationalen Forschungsbeiträgen verwendet wurden, stammen aus dem Themenbereich «Games» und werden in der Programmierlernumgebung «Scratch» umgesetzt. Dieses Resultat stimmt mit den Ergebnissen von Zhang und Nouri (2019) überein, die Scratchprojekte mit den gleichen Kategorien analysiert haben. Für das Programmierkonzept «Variablen» mussten die Kategorien für U1 um die Ausprägung «Mathematik» ergänzt werden. In den meisten Beiträgen wurden für die Umsetzung von Programmieraufträgen die Plattform «Scratch» verwendet. Die formulierte Erwartung, dass mehr Beiträge ausschliesslich Plugged-Aktivitäten beschreiben, wurde durch die Lehrmittel und Forschungsbeiträge der U1 bestätigt. Es konnten sieben Unplugged-Aktivitäten gesammelt werden.

4.7.2 Implikationen

Die Ergebnisse zu U1 zeigen, dass die Box-Analogie trotz möglicher FV oft in Lehrmitteln und Forschungsbeiträgen verwendet wird. Unterschiede zwischen Analogien und Zielkonzept, wie dass in einer Box mehrere Sachen Platz haben, müssen im Unterricht thematisiert werden. (Treagust et al., 1998). In zwei Lehrmitteln wird die Verwendung dieser Analogie eingeordnet und Lehrpersonen werden auf Unterschiede und Gefahren hingewiesen. In anderen Lehrmitteln oder zu anderen Analogien konnten keine solchen didaktischen Hinweise gefunden werden.

U1 hat gezeigt, dass in Lehrmitteln und Forschungsbeiträgen zwei bisher unbekannte Analogien zur Erklärung des Programmierkonzepts «Variablen» verwendet werden. Die Briefumschlags-Analogie hat ähnliche strukturelle Eigenschaften wie die Box-Analogie und veranschaulicht das Speichern von Werten. Diese Wertespeicherung wird in Bezug auf Variablen im Unterricht oft thematisiert (Van Der Werf et al., 2023). Die Box und der Briefumschlag unterscheiden sich hinsichtlich der Dimensionalität. Die Alltagsanalogie dagegen zeigt andere Übereinstimmungen mit den Zielkonzept «Variablen». In den untersuchten Beispielen werden Parallelen zwischen der Veränderlichkeit der Werte einer Variable und dem logischen Wertebereich gezogen. Grover et al. (2019) zeigten, dass Aktivitäten mit ebendiesem Alltagsbezug in Bezug auf das Programmierkonzept «Variablen» zu grösseren Lernfortschritten bei Lernenden führten. Welche Alltagsanalogien das Potenzial haben, beim Erklären von Variablen zu helfen, kann anhand dessen noch nicht beurteilt werden.

Die Analyse der Rollen von Variablen hat gezeigt, dass die Rolle «Gatherer» neben «Fixed Value» und «Stepper» oft in Aktivitäten für junge Lernende vorkommt. Dieser Befund hängt potenziell damit zusammen, dass auf dieser Stufe oft mit einem gamebasierten Ansatz gearbeitet wird (Zhang & Nouri, 2019). Der Punktestand ist eine solche Variable mit der Rolle «Gatherer». Punktestände kommen bei sportlichen Wettkämpfen oder analogen sowie digitalen Spielen ebenfalls vor. Paparo et al. (2021) bestätigen, dass in Scratchprojekten von Kindern oft die Variable «Punktestand» vorkommt. Dies legt die Hypothese nahe, dass die Variable «Punktestand» für Lernende der Zielstufe verständlicher sein könnte als andere Variablen.

Die Sammlung von Unplugged-Aktivitäten in U1 kann als Ideenspeicher für mögliche Unterrichtseinheiten angesehen werden.

5. Untersuchung 2: Experten- und Expertinneninterviews

Untersuchung 2 (U2) zielte darauf ab, die Perspektiven von Experten und Expertinnen aus dem Bereich «Variablen» im Informatikunterricht mit Kindern einzuholen, um unterschiedliche Ansichten berücksichtigen zu können, was im DBR-Prozess zentral ist (Geldreich et al., 2019). Ziel von U2 war es, die Literaturrecherche und die ersten Ergebnisse zu U1 mit Fachpersonen zu besprechen. U2 soll einen Einblick in das Verständnis von einem angemessenen Informatikunterricht für Lernende im Alter von 10–12 Jahren zum Programmierkonzept «Variablen» geben. Angelehnt an U1 wurde dazu FF2 entwickelt, die in Kapitel 5.1 vorgestellt wird. Es folgt die Beschreibung der Stichprobe (Kapitel 5.2), der Untersuchungsmaterialien (Kapitel 5.3), der Prozedur (Kapitel 5.4) und des Ablaufs der Datenauswertung (Kapitel 5.5). Anschliessend werden die Resultate in Bezug zu FF2 gestellt (Kapitel 5.6) und diskutiert (Kapitel 5.7).

5.1 Forschungsfrage und Erwartungen

Die leitende FF zu U3 lautete wie folgt:

FF2: Welche Analogien, Aktivitäten, Rollen und didaktischen Hinweise beschreiben Experten oder Expertinnen für das Programmierkonzept «Variablen» für Lernende im Alter von 10–12 Jahren?

Da U1 und U2 im gleichen Zeitrahmen stattgefunden haben, hatte nicht die gesamte Analyse zu U1 Einfluss auf U2. Die ersten Ergebnisse zu U1 zeigten jedoch, dass die Box-Analogie in den untersuchten Lehrmitteln und Forschungsbeiträgen häufig verwendet wird (Kapitel 4.6.1). Es wurde deshalb erwartet, dass die befragten Personen diese Analogie ebenfalls erwähnen werden. Zu dieser Analogie existieren jedoch auch empirische Befunde, die deren Verwendung infrage stellen (Hermans et al., 2018). Deshalb wurde davon ausgegangen, dass weitere Analogien beschrieben werden. Gemäss der Theorie der «Sematic Wave» (Maton, 2013) und den positiven Befunden bei der Verwendung von Unplugged-Übungen (Grover et al., 2019) wurde erwartet, dass die befragten Personen sowohl Aktivitäten in Programmierumgebungen beschreiben als auch solche, die Variablen in einem realen Kontext zeigen. Es wurde versucht, festzustellen, welche Rollen von Variablen nach Kuittinen und Sajaniemi (2004) in diesen beschriebenen Aktivitäten vorkommen. Dabei galt die gleiche Annahme wie in U1, dass «fixed Value», «Stepper» und «Most-recent-holder» am häufigsten vorkommen (Sajaniemi, 2002)²⁴. Der letzte Teil der FF2 bezieht sich auf didaktische Hinweise, die von den Experten und Expertinnen im Interview genannt wurden. Diese Formulierung ist breit gefasst, um mögliche

²⁴ Beim erstellen dieser Annahmen waren die Ergebnisse der U1 noch nicht klar ausgewertet, weshalb hier diese keinen Einfluss auf die Erwartungen hatten.

Beobachtungen und Erfahrungen der befragten Personen gerade im Bezug auf jungen Lernenden miteinschliessen zu können.

5.2 Stichprobe

Für das Interview wurden vier Personen angefragt, von denen drei zugesagt haben. Die vierte Person konnte aus Kapazitätsgründen nicht teilnehmen. Der Anfragetext kann dem Anhang 10.2.1 entnommen werden. Es wurde kein Ersatz mit ähnlicher Expertise gefunden. Das Interview wurde daher mit drei Expertinnen durchgeführt. Gemäss (Döring & Bortz, 2016, S. 375) muss ersichtlich sein, weshalb den Personen ein Status als Person mit Expertenwissen zugesprochen werden kann. Im Rahmen dieser Untersuchung benötigen die Personen Expertise in Bezug auf die Vermittlung von Programmierkompetenzen an Lernende im Alter von 8–12 Jahren. Im Idealfall verfügen die Personen eine Spezialisierung auf das Programmierkonzept «Variablen». Der Fokus wurde darauf gelegt, dass die Personen Wissen aus dem Forschungsbereich sowie praktische Erfahrungen mit Lernenden haben. Für diese Untersuchung gelten somit Personen als Experte oder Expertin, die entweder einen wissenschaftlichen Artikel zum Programmierkonzept «Variablen» veröffentlicht haben oder über Unterrichtserfahrung im Fach «Medien und Informatik» im Zyklus 2 verfügen.

Expertin 1 und 2

Die Expertinnen 1 und 2 haben unter anderen im Rahmen einer Forschungsarbeit eine umfassende Analyse der bisherigen Literatur rund um das Programmierkonzept «Variablen» erstellt. Aus dieser Forschungsarbeit sind Unterrichtsmaterialien entstanden, die zur Stichprobe von U1 gehören. Expertin 1 arbeitet als Forscherin im Bereich des Mathematik- und Informatikunterrichts in den USA. Expertin 2 arbeitet ebenfalls in den USA, jedoch als Spezialistin für Lernen und Lehrplanentwicklung.

Expertin 3

Expertin 3 hat Praxiserfahrung als Lehrperson im Zyklus 2. Im Rahmen ihrer Masterarbeit hat sie sich explizit mit dem Programmierunterricht in der 5. und 6. Klasse auseinandergesetzt. Die Expertin arbeitet als wissenschaftliche Mitarbeiterin an einer pädagogischen Hochschule in der Schweiz.

5.3 Untersuchungsmaterialien

Um FF2 zu beantworten, wurde ein leitfadengestütztes Interview durchgeführt. Das Interview wurde als Co-Experteninterview umgesetzt, da sich die interviewende Person bereits intensiv mit dem Thema auseinandergesetzt hat. In der Interviewsituation hat daher ein inhaltlicher Austausch zwischen Fachpersonen stattgefunden (Döring & Bortz, 2016, S. 376) Der

Leitfaden für die Interviews wurde selbst konstruiert und kann dem Anhang 10.2.4/10.2.5 entnommen werden. Um FF2 zu beantworten, wurden in einem ersten Teil offene Fragen zu den Themen «Kinder und Variablen», «Informatikunterricht mit Variablen» sowie «Analogien für Variablen im Informatikunterricht» gestellt. Im Teil «Kinder und Variablen» wurde darauf abgezielt, welche Beobachtungen von Lernenden im Umgang mit Variablen gemacht wurden und welches Verständnis über Variablen als angemessen erachtet wurde. Gemäss Döring und Bortz (2016, S. 375) äussern Experten und Expertinnen eher strukturelles Fachwissen anstelle von Praxis- und Handlungswissen. Deshalb wurden die interviewten Personen im Teil «Informatikunterricht mit Variablen» aufgefordert zu beschreiben, wie sie das Programmierkonzept «Variablen» jungen Lernenden erklären und welche Aktivitäten sie einsetzen würden. Im Abschnitt «Analogien für Variablen» wurde erfragt, welche Analogien zur Erklärung des Programmierkonzepts «Variable» genutzt werden und wie der Einsatz selbiger eingeschätzt wird. Der zweite Teil des Interviews stellte ein offenes Brainstorming über ausgewählte Analogien in der Gegenüberstellung mit möglichen FV von Lernenden über Variablen dar. Hier wurden die Expertinnen oder Experten aufgefordert laut mitzudenken. Dieser Prozess kann dabei helfen, einen tieferen Einblick in die Denkprozesse der befragten Personen zu erlangen (Döring & Bortz, 2016, S. 375). Ein weiteres Ziel dieses Teils des Interviews bestand darin, dass die Autorin dieser Arbeit ein tieferes Verständnis für den Zusammenhang zwischen Analogie und FV erlangt. Dafür wurden die Analogien Box, mit Post-Its und Murmeln getrennt, und Whiteboard angeschaut. Die Analogien wurden mittels einer Abbildung illustriert. Im Kontext der FV wurde die Zusammenstellung aus Tabelle 6 gezeigt. Um den Ablauf der Interviews zu strukturieren, wurden Präsentationsfolien vorgelegt, welche im Anhang 10.2.6 zu sehen sind. Auf diesen waren die Themen, Analogien und FV ersichtlich. Der Leitfaden wurde im Rahmen eines Probeinterviews mit einer Mitstudentin getestet. Daraufhin wurden zusätzlich die zuvor genannte thematische Unterteilung in der Präsentation visualisiert.

5.4 Prozedur

Die Personen mit Expertenwissen wurden im September 2023 per E-Mail angeschrieben. Hierbei wurde ihr Einverständnis zur Teilnahme am Interview und der Verwendung der Daten eingeholt (Anhang 10.2.2/10.2.3). Die Interviews wurden im Dezember 2023 online durchgeführt; hierbei erfolgte eine Audio-Aufzeichnung. Ein Interview wurde auf Deutsch gehalten, die anderen beiden auf Englisch. Während der Interviews wurden Notizen zu relevant erscheinenden Aussagen angefertigt. Die Interviews dauerten jeweils durchschnittlich 45 Minuten.

5.5 Ablauf der Datenauswertung

Die Audiodateien der Interviews wurden unmittelbar nach der Durchführung mithilfe der Transkriptionssoftware noScribe transkribiert. Es wurden die Einstellung «precise» und die Sprache

«de» oder «en» gewählt; die sprechenden Personen wurden automatisch erkannt. «Überlappende Sprache», «Zeitmarken» und «Pausen markieren» wurden ebenfalls angewählt. Das Transkript wurde einmalig von der Autorin überarbeitet. Dabei wurde mit den Transkriptionsregeln des inhaltlich-semantischen Transkriptionssystems nach Dresing und Pehl (2018) gearbeitet. Die finalen Transkripte können dem Anhang 10.2.8-10.2.10 entnommen werden. Anhand von FF1 und dem Interviewleitfaden wurden Kategorien deduktiv gebildet. Die Transkripte wurden damit in einem ersten Prozess kodiert. Die Ausprägungen der Kategorien wurden teilweise ebenfalls deduktiv gebildet, beispielsweise für die Rollen von Variablen. Weitestgehend erfolgte die Bildung der Ausprägungen der Kategorien jedoch induktiv anhand der Informationen aus den Transkripten. Es folgte ein zweiter Prozess der Kodierung und Verdichtung des Kodiersystems. Das finale Kategoriensystem kann dem Anhang 10.3.4 entnommen werden.

5.6 Ergebnisse

In diesem Abschnitt werden die Ergebnisse der Befragung der Expertinnen zusammenfassend dargestellt, anhand derer FF2 (Welche Analogien, Aktivitäten, Rollen und didaktische Hinweise beschreiben Experten oder Expertinnen für das Programmierkonzept «Variablen» für Lernende im Alter von 10–12 Jahren?) beantwortet werden soll. Bei der Darstellung der Ergebnisse werden die Bezeichnungen «E1», «E2» und «E3» für die entsprechende Expertin verwendet.

5.6.1 Analogien für Variablen

Zunächst wurde ausgewertet, welche Analogien von den Expertinnen im ersten Teil des Interviews aufgezählt wurden. Die Boxanalogie wurde von allen Expertinnen beschrieben. E3 hat zusätzlich erwähnt, dass sie Zählpunkte oder Perlen verwenden würde, um die Veränderung des Wertes zu illustrieren. E1 und E2 erwähnten die Verwendung der Label-Analogie mithilfe von Post-its, die an eine bestimmte Stelle geklebt würden. E2 ergänzte, dass für die Veranschaulichung von Variablen auch ein Whiteboard oder eine andere abwischbare Tafel verwendet werden könnte. E1 und E3 beschrieben die Verwendung der Briefumschlags-Analogie. Ein Verweis auf eine mögliche Alltagsanalogie wurde von E1 und E2 vorgebracht. Als Beispiele wurden die Uhrzeit, die Temperatur, ein Kilometerzähler und Schulfächer genannt.

Die Analogien wurden in den Interviews nicht nur benannt –teilweise wurde ihre Verwendung zudem kritisch kommentiert. Die genannten Stärken und Schwächen der Analogien für Variablen werden im Folgenden vorgestellt. E2 beschrieb, dass die Form der Box es den Lernenden erschweren würde, zu verstehen, dass nur der aktuelle Wert der Variable gespeichert werden kann.

But that three-dimensionality of it, that fact that the, the typical variables, if they're not an array variable that holds multiple values, that every time you put something in, it wipes away the older thing, that was the challenge there with that analogy. (Expertin 2, Pos. 58)

Die Erkenntnis, dass nur ein Wert in der Box Platz hat, erachtet auch E1 als anspruchsvoll für junge Lernende. Im zweiten Teil des Interviews wurde die Box-Analogie mit Post-its von E1 und E2 mit FV 1 gekoppelt, die besagt, dass eine Variable mehrere alte Werte speichert. E2 erwähnte, dass die Box-Analogie bei FV 6 auch nicht hilfreich sei, da bei einer Kopie des Wertes eine Box auch geleert werden könnte. E1 gab an, zu vermuten, dass die Box-Analogie im Hinblick auf FV 4 helfen könnte, indem die Speicherung des Werts mit der Box visualisiert wird. So könne die Box-Analogie die mentale Vorstellung von Werten bei Lernenden unterstützen.

Laut E1 und E2 kann es Lernenden helfen, wenn die Box-Analogie mit Murmeln verwendet wird, weil dabei die konkrete Vorstellung des Werts gefördert würde. Auch die Veränderung des Werts könne damit gezeigt werden, gab E3 an. Ihr zufolge würde diese Analogie jedoch dadurch geschwächt, dass sie sich auf die Verwendung von Datentypen beschränkt, die mit Zahlen zu tun haben.

Die Whiteboard-Analogie könnte laut allen drei Expertinnen bei FV 1 hilfreich sein, da der Wert zuerst weggeputzt werden muss, bevor ein neuer Wert gespeichert werden kann. E2 gab an, dass auch das Kopieren von Werten (FV 6) damit adäquat gezeigt werden könne. E3 sagte aus, sich vorstellen zu können, dass das Updaten von Variablen – z. B. «verändere die Variable x um 3» – bei dieser Analogie anspruchsvoll sei, weil die Lernenden die Rechnung im Kopf machen müssten. E2 fügte hinzu, dass diese Analogie in der Umsetzung aufgrund des Materials anspruchsvoller sei.

E1 und E2 erklärten, dass sie die Lable-Analogie mit Post-its verwendet hätten, um den Lernenden zu verdeutlichen, dass nur der aktuelle Wert in der Variable gespeichert ist (FV 1).

When what we really want kids to say is, you can only have one thing at a time in there. And if you go to put something else, it, it makes the other thing even disappear or you have to take it out first. And so, that led us to using flat boxes, like a square or a rectangle, and using post-its, right? One post-it at a time. (Expertin 2, Pos. 58)

Die Alltagsanalogie zeige laut E1 die Veränderlichkeit von Variablen. Gleichzeitig könnten die Folgen dieser Veränderung im Alltag beobachtet werden. Die Variablen in den Alltagsbeispielen könnten jedoch teilweise nicht wie Variablen in Programmen von Programmierer:innen gesteuert werden.

Like the problem with those analogies is like they are not. (..)They're not in the position of a programmer in the sense that they'd be in control of what the values are and they are not defining the process that

updates the variable, but at least they're making a connection that there is something that is fluid and they can think about the concept of the time of day and realize it's changing. (Expertin 1, Pos. 30)

Die drei befragten Expertinnen schätzten die Verwendung von Analogien für das Programmierkonzept *Variable* als sinnvoll ein. Sie waren sich auch einig darüber, dass der Einsatz von Analogien didaktisch abgeschätzt und geplant werden muss. Dabei sei es laut E1 entscheidend, auf welche Eigenschaft des Programmierkonzepts *Variable* Lernende hingewiesen werden sollten. Sie erklärte zudem, welche Analogie ihrer Ansicht nach für welche Eigenschaften einer Variable sinnvoll sei:

Yeah, I do. No analogy is perfect, right? So it depends on the part of the the concept you're trying to really hit home. So that's so, for example, like the box is really useful when you're first starting out because it's not really that important that they know right away this idea that you're erasing what was there before when you update. Right. So even just thinking there's something in the box, I can put whatever I want in the box, take it out, put something else in. (.) That's really useful, I think, for the concept, again, having that mental model for a thing that you can put something into, but you don't know what it is yet, the box is closed. So like that's if that's what I'm trying to get to is that basic idea of an unknown, then the box is useful. But again, like the labels might be more useful is when I'm really trying to get at this, like. If I update the variable, the old value is gone. (..) And again, like if I'm trying to get to I was bringing up the odometer and the time and stuff, I'm really trying to get to this dynamic nature of variables, which I think is very hard for kids, something like that. (..) Like the problem with those analogies is like they are not. (..) They're not in the position of a programmer in the sense that they'd be in control of what the values are and they are not defining the process that updates the variable, but at least they're making a connection that there is something that is fluid and they can think about the concept of the time of day and realize it's changing. (Expertin 1, Pos. 30)

Die Expertin vertrat die Ansicht, dass keine Analogie perfekt sei, sondern jede Analogie einen bestimmten Teilaspekt des Programmierkonzepts *Variable* besonders effektiv abbilden könne. E2 schätzte die Box-Analogie beispielsweise für den Beginn des Lernprozesses als sinnvoll ein. Anschliessend würde sie andere Aspekte anhand anderer Analogien erklären. Zusätzlich würde sie im Unterricht gezielt mit Lernenden über Schwierigkeiten, wie eine Variable nicht mehrere Werte speichern kann, sprechen.

5.6.2 Aktivitäten für Variablen

Alle drei Expertinnen wurden gefragt, wie sie eine Unterrichtssequenz mit Lernenden im Alter von 10–12 Jahren zum Programmierkonzept *Variablen* gestalten würden. In den Antworten wurden unterschiedliche Aktivitäten genannt, die Tabelle 11 entnommen werden können. Es wurde zwischen Unplugged-Aktivitäten und solchen unterschieden, für die die Verwendung eines Geräts vorgesehen war. Bei zwei der zehn Aktivitäten ist eine Umsetzung in einer

Programmierlernumgebung notwendig; sechs Aktivitäten können in einem Unplugged-Setting umgesetzt werden. Die zwei verbleibenden Aktivitäten sind in beiden Formen möglich. Die Expertinnen waren sich darin einig, dass ein vielfältiges Angebot an Aktivitäten in einer Programmierumgebung entscheidend ist, um ein Verständnis für das Programmierkonzept «Variable»n aufzubauen.

Tabelle 11

In den Interviews mit den Expertinnen erwähnte Aktivitäten (n = 3)

Name	Beschreibung	E		Rolle (Sajaniemi et al., 2006)
Zahlenrätsel	Eine Rechenaufgabe, bei der am Schluss immer das gleiche Resultat entsteht.	E3	u	unklar
Kontostand	Beim Einkaufen und Verkaufen ändert sich der Kontostand	E3	u	Gatherer
Gericht kochen	Ein Gericht wird auf die Anzahl der Kinder in der Klasse angepasst	E3	u	Fixed Value
Roboter	Es wird eine Anleitung für einen Roboter beschrieben, der die Fläche oder den Umfang eines Rechtecks berechnet. Die Eingabewerte werden immer wieder verändert.	E1 E2	u	Fixed Value (Länge und Breite) Gatherer (Fläche/ Umfang)
Input-Output-Maschine	Es gibt einen Startwert und einen Endwert und die Lernenden müssen herausfinden, was mit dem Wert passiert sein könnte.	E1	u	Gatherer
Gedichte/Lieder	Lieder anhören, bei denen sich nur ein Teil jeweils ändert. Eigenes Beispiel: If you're happy and you know it clap your hands. Diese Lieder können gesungen oder auch in Scratch programmiert werden.	E2	u/p	Most-recent-holder
Spiel programmieren	Die Kinder müssen ein Spiel mit der Variable «Punktestand» programmieren	E1	p	Gatherer
Codebeispiele anschauen	Im Beispiel müssen Variablen gesucht und ggf. auch verändert werden. Diese Codebeispiele können sie selbst verändern.	E1	u/p	unklar

Debugging	Ein Beispiel zeigen, bei dem eine Zahl mehrfach verwendet wird. Diese wird nun verändert.	E1	p	Fixed Value
Verrückte Lücke	Spiel, bei dem zuerst Wörter zu verschiedenen Kategorien gesucht werden. Anschliessend wird ein Text gelesen, in dem die gesuchten Wörter in die Lücken eingefügt werden.	E2	u	Fixed Value

Anmerkung: u = unplugged, p = plugged.

Bei den beschriebenen Aktivitäten wurde untersucht, welche Rollen von Variablen nach Kuittinen und Sajaniemi (2004) vorkommen. Im Fall von zwei Aktivitäten konnte auf Grund der einfachen Beschreibung nicht eruiert werden, welche Rolle die Variable einnimmt. Von den acht beschriebenen Rollen von Variablen kamen bei den von den Expertinnen thematisierten Aktivitäten die Rolle «Fixed Value», «Gatherer» und «Most-recent-holder» vor. Die Rolle «Gatherer» wird für die Variablen «Kontostand», «Fläche oder Umfang eines Rechtecks» und «Punktstand» verwendet. Die Expertinnen haben die verschiedenen Rollen von Variablen nicht explizit erwähnt. E3 hat jedoch zuerst das Beispiel mit dem Kontostand beschrieben und anschliessend angegeben, dass eine weitere Übung relevant sei, bei der nicht der konkrete Wert im Mittelpunkt stünde, sondern etwas berechnet würde. In diesem Zusammenhang wurden die verschiedenen Rollen von Variablen angesprochen. Das Konzept war E3 jedoch nicht bekannt.

5.6.3 Didaktische Hinweise für Variablen

In den Antworten der Expertinnen wurden didaktische Hinweise zum Programmierkonzept «Variablen» in unterschiedlichen Kategorien gefunden.

Herausforderungen des Programmierkonzepts «Variablen»

Die Expertinnen haben im Kontext verschiedener Fragen beschrieben, inwiefern das Programmierkonzept speziell für Lernende im Alter von 10–12 Jahren herausfordernd ist. E3 hat Variablen als komplexeres Konzept beschrieben, die das Feld des Programmierens öffnen und somit anspruchsvollere Projekte ermöglichen würden. Sie gab zudem an, dass für Variablen bei den Lernenden ein inneres Vorstellungsvermögen – eine Art Zwischenspeicher – erforderlich sei, um die Werte zu verstehen. Die drei Expertinnen waren sich darin einig, dass Lernende in diesem Alter verstehen könnten, dass Variablen etwas sind, das unbekannt ist. E1 sei es gelungen, jungen Lernenden beizubringen, dass Variablen veränderbar sind. Wie im Kontext der Aktivitäten in der Input-Output-Maschine beschrieben, hätten ihre Lernenden verstanden, dass eine Veränderung eines Werts stattfinden kann. Ebendiese Veränderlichkeit der Werte von Variablen unterscheidet die informatische Nutzung von Variablen von der

mathematischen. Als anspruchsvolle Eigenschaften von Variablen wurden von allen drei Expertinnen die Initialisierung, das Aktualisieren und das Verändern von Variablen beschrieben. Beispielsweise gibt es gemäss E1 einen klaren Unterschied zwischen dem Auftrag die Variable Länge ist neu 4 und dem Auftrag die Variable Länge ist um eins grösser. Der zweite Auftrag ist für Lernende in diesem Alter anspruchsvoll, wie hier beschrieben ist:

So instead of telling them replace four with five, we say increase the variable value by one. That's harder for them, I think, because they're thinking about the process of updating and what the final result is and don't quite. (.) Put together that they need the process to know what the you know, it's it's almost like they have trouble coordinating those two processes at the same time. (Expertin 1, Pos. 16)

Relevanz des Vorwissens

E1 und E2 haben mehrfach das Vorwissen oder auch den Lebensweltbezug als zentralen Aspekt für Unterrichtsvorbereitung genannt. Gemäss den beiden Expertinnen ist es entscheidend, zu wissen, wo bei den Lernenden angeknüpft werden kann. «So, keeping, paying close attention to the first context in which children may encounter variables, I think it's extremely important for contemporary education on variables» (Expertin 2, Pos. 18). In den Interviews wurden für potenzielle Anknüpfungspunkte Bezüge zur Mathematik hergestellt und Onlinegames sowie die beschriebenen Alltagsanalogien genannt.

Sensibler Gebrauch von Sprache für Variablen

E1 und E2 erwähnten in den Interviews, dass das Sprechen über Variablen anspruchsvoll sei. Das gilt laut den beiden sowohl für Lehrpersonen als auch für die Lernenden. Dabei habe E2 die Erfahrung gemacht, dass solche Überlegungen besonders relevant seien, wenn beispielsweise Unterrichtsmaterial übersetzt wird. E2 ist überzeugt, dass es für diese Arbeit entscheidend sei, sich zu überlegen, mit welchen Wörtern das Programmierkonzept «Variable» erklärt werden kann, ohne den Begriff «Variable» zu erwähnen.

Weitere Kategorien und Aussagen der Expertinnen zielen auf den Programmierunterricht im Allgemeinen ab. Es wird an dieser Stelle darauf verzichtet, diese zu präsentieren, da sie für die Beantwortung von FF2 nicht relevant sind.

5.7 Zwischendiskussion und Fazit

In diesem Kapitel wird FF2 beantwortet und erläutert, inwieweit die Ergebnisse für die vorliegende Arbeit relevant sind.

5.7.1 Beantwortung von FF2

Um FF2 «Welche Analogien, Aktivitäten, Rollen und didaktischen Hinweise beschreiben Experten oder Expertinnen für das Programmierkonzept «Variablen» für Lernende im Alter von 10–12 Jahren?» zu beantworten, wurden Expertinnen in einem Leitfadeninterview befragt. Die Aussagen der Expertinnen können im Hinblick auf FF2 in vier Teilantworten gegliedert werden:

- Die Expertinnen beschrieben die Box-Analogie, die Label-Analogie, die Briefumschlags-Analogie, die Whiteboard-Analogie und die Alltagsanalogie. Wie erwartet, wurde die Box-Analogie von allen Expertinnen beschrieben. Die Beschreibung, dass die Box-Analogie wertvoll sei, um den Lernenden den Aspekt des Unbekannten aufzuzeigen, deckt sich mit den Ergebnissen von Hermans et al. (2018). Die Ergebnisse aus U2 und die Ergebnisse der Studie von Hermans et al. (2018) überschneiden sich hinsichtlich der Erkenntnis, dass die Box-Analogie zu Beginn des Lernprozesses hilfreich sein kann, jedoch zu Schwierigkeiten führen kann, sobald die Programmieraufgaben komplexer werden. Nicht bereits aus der Literatur bekannt sind dabei die Briefumschlags-Analogie und die Alltagsanalogie. Diese beiden Analogien wurden auch in U1 in Lehrmitteln und Forschungsbeiträgen gefunden. Variablen weisen unterschiedliche Eigenschaften auf, die gemäss den Expertinnen nicht mit einer einzelnen Analogie erklärt werden können. In Bezug auf Analogien weisen die Aussagen der Expertinnen darauf hin, dass keine bisher bekannte Analogie für das Programmierkonzept «Variablen» ideal ist. Gemäss den Bedingungen für Analogien von Kircher und Girwidz (2020) ist eine gewisse Abweichung von Analogien vom Zielkonzept unproblematisch, muss jedoch zwingend thematisiert werden. Eine Expertin hat vorgeschlagen, Analogien zu verwenden, um bestimmte Eigenschaften von Variablen zu erklären.
- Die Expertinnen haben zehn Aktivitäten beschrieben, von denen sechs zu den Unplugged-Aktivitäten, zwei zu den Plugged-Aktivitäten und zwei weitere zu beiden Kategorien gezählt werden konnten. Dieses Resultat entspricht der Erwartung, dass die Expertinnen mehr Aktivitäten beschrieben, die Variablen in einem Unplugged-Setting zeigen. Die Relevanz solcher Unplugged-Aufgaben im Informatikunterricht wird in diversen Studien bestätigt (Brackmann et al., 2017; del Olmo-Muñoz et al., 2020; Munasinghe et al., 2023; Sun et al., 2021).
- In den von den Expertinnen beschriebenen Aktivitäten konnten Variablen mit den Rollen «Fixed Value», «Gatherer» und «Most-recent-holder» gefunden werden. «Fixed Value» und «Most-recent-holder» decken sich dabei mit den von Sajaniemi (2002) am häufigsten gefunden Rollen von Variablen in Programmierlehrmitteln. «Fixed Value» und «Gatherer»

decken sich dabei mit den Rollen, die in den Lehrmitteln und Forschungsbeiträgen aus U1 vorgekommen sind.

- Im Kontext der didaktischen Hinweise haben die Expertinnen in U3 verschiedene Schwierigkeiten mit dem Programmierkonzept «Variablen» erwähnt. Verschiedene Forschende beschreiben ebenfalls, dass das Programmierkonzept «Variablen» anspruchsvoll für Lernende ist (Grover et al., 2019; Hermans et al., 2018; Kohn, 2017). Weiter wurde von den Expertinnen die Relevanz des Vorwissens der Lernenden bei der Aufbereitung von Unterrichtsmaterial betont. Dieses Vorwissen nimmt auch eine zentrale Rolle im Modell der Didaktischen Rekonstruktion von Diethelm et al. (2011) ein. Zuletzt wurde von den Expertinnen ein sensibler Gebrauch von Sprache über Variablen im Unterricht gefordert. In der Theorie der «Semantic Wave» von Maton (2013) steht ebenfalls die Frage im Fokus, wie komplexe Konzepte erklärbar gemacht werden können. Das Modell unterstützt in der Phase des Erschliessens die Verwendung von Analogien, was auch für den Informatikunterricht gilt (Waite et al., 2019).

5.7.2 Implikationen

Die Ergebnisse aus U2 zeigen, dass keine Empfehlung zur Verwendung einer einzigen Analogie ausgesprochen wurde. Von den Expertinnen wurde dazu geraten, Analogien für die Veranschaulichung einer bzw. weniger Eigenschaften von Variablen zu verwenden. Es gibt bereits einige solcher spezifischen Analogien. Die Parkplatz-Analogie für Variablen (Pearce, 2024) verfolgt beispielsweise das Ziel zu erklären, welche Rolle der Datentyp von Variablen spielt. Die Onlinesammlung von Notional Machines für informatische Konzepte beschreibt in der Erklärung der Analogien, auf was durch die Analogie aufmerksam gemacht wird, wann die Analogie verwendet werden soll und welche Gefahren beachtet werden müssen (*Analogy-Based Notional Machines*, o. J.). Aubusson et al. (2006) unterstützen eine genaue Aufbereitung von Analogien.

In Bezug auf die Rollen von Variablen kann festgehalten werden, dass die Hälfte der Rollen nicht in den Aktivitäten vorkommen, die in den Lehrmitteln und Forschungsbeiträgen (U1) oder von den Expertinnen (U2) erwähnt wurden, und auch bei untersuchten Programmierlehrmitteln für prozedurale Programmiersprachen seltener sind als andere Rollen (Sajaniemi, 2002). Das deutet darauf hin, dass die Rollen «Fixed Value», «Stepper», «Gatherer» und «Most-recent holder» eine grössere Bedeutung für junge Lernende haben könnten. Die Variable Punktestand wurde auch in den beschriebenen Aktivitäten der U2 gefunden.

U1 hat gezeigt, dass in Lehrmitteln und Forschungsbeiträgen zur Erklärung vom Programmierkonzept «Variablen» neben Aufgaben in Programmierlernumgebungen nicht immer auch Unplugged-Aktivitäten verwendet werden. Die Expertinnen empfehlen jedoch den Einsatz solcher

Aufgaben in Kombination mit Programmieraufgaben. Die in Tabelle 11 aufgelisteten Aktivitäten – unplugged und plugged – können als Ideenpool angesehen werden, auf den bei der Entwicklung von Unterrichtssequenzen zum Programmierkonzept «Variablen» zurückgegriffen werden kann.

Um Lernenden das Programmierkonzept «Variablen» zu erklären, ist es laut den Expertinnen entscheidend, an ihr Vorwissen anzuknüpfen. Die Vorstellungen von Lernenden nehmen auch im Modell der Didaktischen Rekonstruktion eine zentrale Rolle ein und beeinflussen die Strukturierung von Informatikunterricht (Diethelm et al., 2011). Im Bereich der Variablen sind mögliche FV bekannt (Kapitel 2.3.1), über intuitive oder elaborierte Vorstellungen von Lernenden in diesem Alter liegen nur wenige Informationen vor. U1 und U2 haben gezeigt, dass es möglich sein könnte, Variablen aus dem Alltag als Analogien für Variablen zu verwenden. Es gilt nun herauszufinden, welche Alltagsanalogien sich dafür eignen und welche Eigenschaften von Variablen damit erklärbar gemacht werden können. Dabei ist nicht die Frage relevant, welche Übereinstimmungen es im Allgemeinen gibt, sondern, vielmehr welche Übereinstimmungen Lernende im Alter von 10–12 Jahren zwischen der Analogie und dem Programmierkonzept «Variablen» beschreiben.

6. Untersuchung 3: Kinderinterviews

In der heutigen digitalen Welt ist das Verständnis grundlegender Programmierkonzepte für Lernende von zentraler Bedeutung (Grover & Pea, 2018). Insbesondere das Konzept der Variablen, das in zahlreichen Programmiersprachen eine Schlüsselrolle spielt, stellt eine Herausforderung für junge Lernende dar (Grover et al., 2019; Hermans et al., 2018; Kohn, 2017). Analogien sind ein möglicher Erklärungsansatz für Variablen und sollen dabei helfen, den Ablauf eines Teils eines Programmes zu visualisieren (Hermans, 2019). U1 und U2 haben gezeigt, dass unter anderem Alltagsanalogien, wie Temperatur oder Zeit, hierbei nützlich sein könnten. Es ist jedoch noch nicht klar, inwiefern Lernende im Alter von 10–12 Jahren eine Übereinstimmung zwischen dem Alltagskonzept und dem Programmierkonzept «Variablen» erkennen. Gemäss Aubusson (2006) müssen mögliche Analogien vor dem Einsatz genau analysiert werden. Untersuchung 3 (U3) soll also Aufschluss darüber geben, wie Kinder im Alter von 10–12 Jahren diese Alltagskonzepte beschreiben und ob Überschneidungen mit dem Programmierkonzept «Variablen» oder einzelnen Eigenschaften davon erkennbar sind.

U3 soll nicht nur mögliche Lebensweltbezüge von Lernenden aufzeigen, sondern auch die Vorstellungen von Lernenden zum Programmierkonzept «Variablen» beleuchten. In der Didaktischen Rekonstruktion gemäss Diethelm et al. (2011) sollen diese Vorstellungen einen grundlegenden Einfluss bei der Entwicklung von Informatikunterricht haben. Laut Kattmann et al. (1997) sind sie ebenso relevant wie das fachliche Wissen, das vermittelt werden soll. Die Vorstellungen von Lernenden im Bereich der Informatik können in folgende Kategorien eingeteilt werden: intuitive und elaborierte Vorstellungen sowie FV (Babari et al., 2023). FV über Variablen sind bereits mehrfach erforscht worden (Kapitel 2.3.1). Ein weiteres Ziel von U3 besteht darin, herauszufinden, welche intuitiven oder potenziell bereits elaborierten Vorstellungen Lernende zum Programmierkonzept «Variablen» haben.

Diese beiden Ziele der U3 dienen der genauen Untersuchung von verschiedenen Beschreibungen von Lernenden. Das Erkenntnisinteresse dieser explorativen qualitativen Studie liegt in der Generierung möglicher Hypothesen für adäquate Erklärungsansätze zum Programmierkonzept «Variablen».

6.1 Forschungsstand zur Vorstellung von Alltagsvariablen

Neben der Zeit und der Temperatur hat sich in den Gesprächen mit den Expertinnen auch der Kilometerzähler als mögliche Alltagsanalogie herauskristallisiert (Kapitel 5.6.1). Der Kilometerzähler wurde durch die Alltagsvariable Geschwindigkeit ersetzt. Dieser Entscheid wird dadurch begründet, dass die Geschwindigkeit im Mathematikunterricht der 5. und 6. Klasse

vorkommen kann (Affolter et al., 2018) und Kinder deshalb eher Erfahrungen darüber gesammelt haben.

Die Grundlage für wissenschaftliche Studien ist der aktuelle Forschungshintergrund (Döring & Bortz, 2016, S. 158), weshalb eine Recherche zu bereits bekannten Vorstellungen von Lernenden zu den Alltagsvariablen «Zeit», «Temperatur» und «Geschwindigkeit» durchgeführt wurde. Die Resultate werden in diesem Kapitel präsentiert.

6.1.1 Temperatur

Phänomene wie Temperatur können Kinder bereits im Vorschulalter beobachten und erleben (Steffensky, 2018). In einer Studie mit Kindergartenkindern zu Hitze und Temperatur wurde untersucht, wie Kinder die Erwärmung von Wasser in der Sonne im Vergleich zu Wasser im Schatten verstehen. Die Ergebnisse zeigen, dass es den Kindern leichter fiel, korrekte Vorhersagen über Temperaturveränderungen zu treffen, als diese zu erklären (Pahl et al., 2022). Diese Antworten zeigen, dass bereits junge Kinder dazu in der Lage sind, Veränderungen der Temperatur vorherzusagen. Lernende teilen unterschiedliche Temperaturen oft in die Kategorien «kalt», «mittel» und «heiss» ein (Choi et al., 2001). Eine Studie aus Korea zeigte, dass 4- bis 11-jährige Kinder dazu neigen, Temperatur als summierende Grösse zu betrachten. Das würde bedeuten, dass sich die Temperatur von zwei Flüssigkeiten addiert, wenn diese vermischt werden (Paik et al., 2007). Dieser Befund zeigt eine gewisse Übereinstimmung mit FV 3 (vgl. Kapitel 2.3.1), bei der Lernende Variablen ebenfalls als summierendes Konstrukt sehen. Lernende konstruieren ihre Vorstellung über Temperatur durch Beobachtungen in ihrem Alltag. Diese können gegensätzlich zu wissenschaftlicher Gesetzmässigkeiten sein (Koh & Paik, 2002). Lernende ordnen teilweise bestimmten Materialien die Eigenschaft «warm» (z. B. Stoff) oder «kalt» (z. B. Metalle) zu. Die Begriffe «Temperatur» und «Wärme» werden zudem oft synonym verwendet und eng an körperliche Empfindungen geknüpft. (Fischler & Schecker, 2018).

Schweizer Kinder lernen gemäss dem Fachlehrplan NMG des Lehrplans 21 (D-EDK, 2016b), Beobachtungen und Messungen zu Temperatur durchzuführen und diese Ergebnisse einzuordnen sowie darzustellen²⁵.

6.1.2 Zeit

Das Verständnis von Zeit ist komplex und vielschichtig, da es mathematische, sprachliche, logische und historische Dimensionen umfasst (Stow, 1999). In der Forschung besteht Uneinigkeit darüber, welche dieser Dimensionen den grössten Einfluss auf das Verständnis von Zeit hat. Friedman (1982) betont die Bedeutung des mathematischen Verständnisses für die

²⁵ LP21 Kompetenz Wetter: NMG.4.4.1e / Eigenschaften von Stoffen: NMG.3.3.2e

zeitliche Entwicklung, während Hoodless (2002) hervorhebt, dass die sprachliche Dimension für das Verständnis von Zeit bei Kindern entscheidend ist. Stow (1999) ergänzt, dass die sprachliche Dimension die sichtbarsten Hinweise auf das Zeitverständnis von Kindern liefert. Eine Studie mit 6–9-jährigen Kindern zeigte, dass jüngere Kinder Zeitbegriffe wie «alt», «modern» oder «neu» zum Beschreiben von Bildern verwendeten, während ältere präzisere Begriffe wie «neuer als» oder «älter als» verwendeten (Solé, 2019). Zhang und Hudson (2018) fanden heraus, dass Vorschulkinder vergangene Ereignisse besser verstehen als zukünftige. Ihre Forschung zeigt auch, dass das Verständnis von Zeit ein langsamer und komplexer Prozess ist, der von der Abstraktheit des Zeitbegriffs geprägt ist: Zeit kann nicht gesehen, gehört, gerochen oder berührt werden.

Im Lehrplan 21 sind Aspekte zum Thema «Zeit» ebenfalls in zwei unterschiedlichen Fachbereichen lokalisiert. Dabei sollen Lernende im Fach NMG einerseits ein Verständnis vom Zeitbegriff aufbauen und selbiges beispielsweise in einem Zeitstrahl anwenden. Darin ist auch enthalten, dass Lernende im Zyklus 1 lernen, die Uhrzeit zu lesen²⁶. Andererseits begegnen die Lernenden in der Mathematik der Masseinheit «Zeit» und stellen damit Berechnungen an²⁷.

6.1.3 Geschwindigkeit

Geschwindigkeit gilt als physikalische Grösse, die beschreibt, wie schnell und in welche Richtung sich ein Körper im Laufe der Zeit bewegt, und wird in Metern pro Sekunde (m/s) oder Kilometern pro Stunde (km/h) angegeben («Geschwindigkeit», 2024). Schecker und Willhelm (2018) haben Vorstellungen von Lernenden zu Geschwindigkeit gesammelt. Lernende betrachten Geschwindigkeit oft als Mass für die Schnelligkeit eines Körpers; die Richtung wird dabei häufig nicht berücksichtigt. Lernende erkennen Beschleunigung als Änderung der Geschwindigkeit und verstehen darunter ein Schneller- oder Langsamerwerden.

Schweizer Kinder lernen gemäss dem Fachlehrplan Mathematik des Lehrplan 21 (QUELLE) Geschwindigkeit anhand von Wertetabellen²⁸ und Alltagssituationen wie der Geschwindigkeit eines Autos kennen²⁹.

Zusammenfassend sind die drei Alltagsvariablen «Temperatur», «Zeit» und «Geschwindigkeit» Teil der Kompetenzstufen des Lehrplans 21 und stellen Phänomene dar, die bereits junge Kinder im Alltag beobachten.

²⁶ LP21 Kompetenz NMG.9.1

²⁷ LP21 Kompetenz MA3.A.1.2h

²⁸ LP21 Kompetenz MA.3.A.3.2e

²⁹ LP21 Kompetenz MA.3.C.2.f

6.2 Forschungsfragen

Die erste leitende FF zu U3 lautet wie folgt:

FF3: Welche Übereinstimmungen lassen sich zwischen dem Programmierkonzept «Variablen» und den Beschreibungen der Alltagsvariablen Zeit, Temperatur und Geschwindigkeit von Lernenden im Alter von 10-12 Jahren feststellen?

Die Erkenntnisse aus dieser Fragestellung sollen es ermöglichen zu verstehen, ob und welche dieser Alltagsanalogien sich für die Erklärung von Variablen bei jungen Lernenden eignen könnte. Dazu werden die formulierten Beschreibungen der Lernenden mit den in Kapitel 2.1.4 beschriebenen Eigenschaften verglichen. Aus den bisherigen Forschungsergebnissen über Vorstellungen von Lernenden zu diesen Alltagsvariablen konnten nur wenige Bezüge hergestellt werden. Aus diesem Grund wurden keine Erwartungen für diese FF3 formuliert.

Die zweite leitende FF lautet:

FF4: Wie beschreiben Lernende im Alter von 10–12 Jahren das Programmierkonzept «Variablen», nachdem sie verschiedene Scratchprogramme mit Variablen angeschaut haben?

Neben der Annahme von Rich et al. (2022), dass Lernende sich zuerst mit Variablen aus der Sicht des Computers auseinandersetzen und sich erst in einem zweiten Schritt in die Lage einer programmierenden Person versetzen können, existieren kaum Befunde über die Beschreibungen Lernender zu Variablen. FF4 soll näher beleuchten, welche Vorstellungen Lernende hinsichtlich des Programmierkonzepts «Variablen» beschreiben, nachdem sie dieses im Kontext von Scratchprogrammen kennengelernt haben. Da das Programmierkonzept «Variablen» im Lehrplan 21 erst im Zyklus 3 beschrieben wird, wird erwartet, dass die Lernenden grösstenteils vereinfachte, intuitive Beschreibungen über das Programmierkonzept «Variablen» machen werden.

6.3 Stichprobe

Um FF3 und FF4 zu beantworten, wurden Lernende der 6. Klasse in Form eines Leitfadenterviews befragt. Bei dieser Art von qualitativen Studien werden oft Stichproben verwendet, die sich im ein- bis zweistelligen Bereich bewegen, damit der Schwerpunkt auf die detaillierte und interpretative Rekonstruktion der Fälle gelegt werden kann. Für die Auswahl der Stichprobe hat sich im Bereich der qualitativen Studien die absichtsvolle Auswahl von Fällen

durchgesetzt (Döring & Bortz, 2016, S. 302). In der vorliegenden Arbeit wurden 10 Kinder aus zwei Klassen ausgewählt. Gemäss Reinders (2016) steigt die Länge der Antworten der Kinder mit ihrem Alter. Solche längeren Antworten erhöhen die Wahrscheinlichkeit, die Perspektive der Kinder zu verstehen. Aus diesem Grund wurden für diese Untersuchung nur 6. Klassen angefragt. Beide Klassen stammen aus ländlich geprägten Regionen der Schweiz. Bedingung für die Teilnahme der Klasse war, dass die Lehrpersonen mit den Kindern bereits mit Scratch gearbeitet haben. Die Klassenlehrpersonen wurden per E-Mail angefragt. In der Anfrage wurden sie dazu aufgefordert, Auskunft über die Scratchkenntnisse der Klasse abzugeben. In Klasse 1 hat die Lehrperson den Lernenden einführende Anleitungen abgegeben und anschliessend weiterführende Wahlangebote thematisiert. Nach eigenen Aussagen hat die Lehrperson die Programmierkonzepte «Schleife» und «Bedingung» explizit mit der Klasse besprochen. In Klasse 2 hat die Lehrperson mit der Klasse verschiedene Projekte mit Scratch erstellt. Beispielsweise haben die Kinder im Mathematikunterricht mit Scratch geometrische Muster gezeichnet. Die Klasse nimmt in diesem Jahr zusätzlich an einem Programmierwettbewerb teil. Im Interview selbst wurden die Lernenden gefragt, ob sie neben der Schule weitere Programmiererfahrungen gesammelt haben. Drei Kinder haben angegeben, in der Freizeit regelmässig unterschiedliche Projekte in Scratch zu programmieren. Diese drei Kinder stammen aus der gleichen Klasse. Drei weitere Kinder haben ausgesagt, dass sie früher in der Freizeit programmiert haben oder selten mit Scratch programmieren und vier Kinder programmieren in der Freizeit nicht.

Die Lehrpersonen haben den Kindern ein Informationsschreiben inklusive Einverständniserklärung für die Eltern mitgegeben und kann dem Anhang 10.3.1 entnommen werden. Aus den verbliebenen Lernenden wurden aus jeder Klasse je fünf Kinder ausgewählt. Diese Art der Stichprobenauswahl wird als «Gelegenheitsstichprobe» bezeichnet (Döring & Bortz, 2016, S. 305–307). Die ausgewählten Kinder waren zwischen 11 und 13 Jahre alt. Die relevantesten Angaben zur Stichprobe können Tabelle 12 entnommen werden. Ein detailliertere Beschrieb der Stichprobe kann dem Anhang 10.3.4 entnommen werden.

Tabelle 12

Stichprobenbeschreibung nach Klasse und Geschlecht (n = 10)

Klassen	Klasse 1		Klasse 2	
Geschlecht	männlich	weiblich	männlich	weiblich
Anzahl	3	2	4	1

6.4 Untersuchungsmaterial

Um FF3 und FF4 zu beantworten, wird ein leitfadengestütztes Interview mit fokussiertem Objekt durchgeführt. Der Aufbau und die Erarbeitungsschritte des Interviewleitfadens werden im Folgenden erläutert.

6.4.1 Aufbau des Interviewleitfadens

Zu Beginn des Interviews soll der Gesprächsrahmen genau geklärt werden. Das ist gemäss Vogl (2021) besonders relevant für eine Interviewsequenz mit Kindern. Wehr (2014) zufolge gelten in der Forschung mit Kindern selbige als Experten oder Expertinnen ihrer Lebenswelt. Deshalb ist zu Beginn des Leitfadens eine Einleitung zu finden. Darin wird versucht, den Kindern zu erklären, dass es keine richtigen und falschen Antworten gibt und sie als Experte oder Expertin in dieser Situation gelten. Diese Rolle soll zudem durch einen freundlichen, ermutigenden und geduldigen Umgang mit den interviewten Kindern gestärkt werden (Wehr, 2014).

Der Leitfaden soll aus mehreren Erzählaufforderungen bestehen (Helfferich, 2014). Deshalb wurde entsprechend FF3 zu den drei Alltagsbereichen «Zeit», «Temperatur» und «Geschwindigkeit» je eine offene Erzählaufforderung formuliert. Neben der Erzählaufforderung sind bei Interviews mit Kindern gezielte Nachfragen entscheidend (Vogl, 2021). Der Aufbau des Leitfadens wurde deshalb in die Bereiche «Erzählaufforderung», «Subfragen» beziehungsweise «Nachfragen», «Fragen zur Begründung» sowie «Aufrechterhaltungs- und Steuerungsfragen» eingeteilt. Der letzte Bereich beinhaltet Fragen, die für Kinder gedacht sind, die wenig erzählen und schüchtern sind.

Im zweiten Teil jedes Themenbereichs durften die Kinder jeweils zwei unterschiedliche Scratchprojekte bedienen, die inhaltlich zum dazugehörigen Themenbereich passen. Während dieser Phase wurden die Lernenden dazu aufgefordert, laut zu denken und ihre Gedanken in Worte zu fassen. Diese Methode wird auch «Think-Aloud» genannt (Döring & Bortz, 2016, S. 370–371) und soll dabei helfen, eine tiefere Einsicht in die Überlegungen der Kinder zu erlangen.

Den Teilnehmenden wurde eine randomisierte Reihenfolge der Themenbereiche zugeordnet. Damit soll ein möglicher Reihenfolgeeffekt vermieden werden, der einen potenziellen Einfluss der Reihenfolge der präsentierten Themen auf die Antworten der Teilnehmenden beschreibt (Döring & Bortz, 2016, S. 254–255).

Im letzten Teil des Interviews wurde FF4 fokussiert und darauf abgezielt, dass die befragten Kinder den Begriff «Variablen» in eigenen Worten beschreiben. Das Fokussieren eines Objektes in einem Interview kann dabei helfen, die Sichtweise der befragten Personen offenzulegen (Döring & Bortz, 2016, S. 378). Deshalb wird erst nach der Betrachtung verschiedener

Scratchprogramme mit sichtbaren Variablen mit den Lernenden darüber gesprochen, was sie unter dem Programmierkonzept ›Variablen‹ verstehen. Im ersten Teil des Interviews wird der Begriff ›Variablen‹ von der interviewenden Person nicht verwendet. Tabelle 13 kann ein möglicher Aufbau eines Interviews entnommen werden.

Tabelle 13

Theoretischer Aufbau des Interviewleitfadens

	Teil 1	Teil 2	
Bereich ›Temperatur‹	Erzählaufforderung	Alltagsbeispiel	Spiel
Bereich ›Geschwindigkeit‹	Erzählaufforderung	Alltagsbeispiel	Spiel
Bereich ›Zeit‹	Erzählaufforderung	Alltagsbeispiel	Spiel
Bereich ›Variable‹	Erzählaufforderung		

6.4.2 Scratchprogramme

In diesem Interview wurden zu jedem der drei Teilbereiche zwei Scratchprogramme entwickelt, die den teilnehmenden Kindern als fokussierte Objekte präsentiert wurden. Dies sollten den Lernenden eine Begegnung mit dem Programmierkonzept ›Variablen‹ ermöglichen. Diese Programme wurden mit dem Ziel entwickelt, einen möglichst nahen Bezug zum Lebensalltag der Kinder herzustellen. Dabei wurde auf die im theoretischen Hintergrund beschriebenen Erkenntnisse zu Vorstellungen von Temperatur, Zeit und Geschwindigkeit aus dem Kapitel 6.1 sowie auf die persönliche Unterrichtserfahrung der Autorin zurückgegriffen.

Beim ersten Programm handelte es sich um eine Animation eines möglichst alltäglichen Ablaufs, in dem eine Variabel zu erkennen war. Beim zweiten Programm handelte es sich um ein Spiel, das jeweils eine bis zwei sichtbare Variablen enthielt. Eine der sichtbaren Variablen war jeweils Zeit, Temperatur oder Geschwindigkeit, um einen Bezug zum vorangehenden Gespräch über diese Themen herzustellen. Die zweite Variable, wenn vorhanden, zeigte einen Punktestand. Der Einsatz wird durch die Ergebnisse der U1 und U2 gestützt. Die in den entwickelten Programmen sichtbaren Variablen wurden auf ihre Verwendung hin analysiert. Dabei wurde festgestellt, dass die Rollen ›Gatherer‹ und ›Most-recent-holder‹ häufig auftreten, die Rolle ›Follower‹ zweimal auftritt und die Rolle ›Stepper‹ einmal. Dieses Auftreten deckt sich bis auf die Rolle ›Follower‹ mit den Erkenntnissen aus U1, U2 sowie der Studie von

Sajaniemi (2002). Die Rolle «Follower» musste aus technischen Gründen eingebaut werden und übernimmt jeweils den Wert einer Variable mit der Rolle «Most-recent-holder». Auf die Rolle «Fixed Value», die eigentlich häufig von Programmieranfängern genutzt wird, wurde verzichtet. Begründet wird dieser Entscheid damit, dass bei den sichtbaren Variablen die Veränderlichkeit der Werte sichtbar gemacht werden wollte. Eine Übersicht zu den Rollen von Variablen in den Scratchprogrammen kann Tabelle 14 entnommen werden.

Tabelle 14

Analyse der Rollen von Variablen in den entwickelten Scratchprogrammen

Bereich	Alltagsprogramm	Spiel
Temperatur	Most-recent-holder	Temperatur: Follower Abstand*: Most-recent-holder Punktestand: Gatherer
Geschwindigkeit	Most-recent-holder	Geschwindigkeit: Most-recent-holder Punktestand: Gatherer
Zeit	Uhrzeit: Follower Antwort*: Most-recent-holder	Timer: Stepper

Anmerkung. *= nicht sichtbare Variable

An dieser Stelle wird beispielhaft eines der sechs Programme vorgestellt. Eine Beschreibung sowie die Verlinkungen der anderen Programme sind dem Anhang 10.3.3 zu entnehmen.

Spiel «Temperatur»

Das Spiel zum Konzept «Temperatur» ist an das bekannte Versteckspiel «Heiss-Kalt» angelehnt, das beispielsweise an Ostern gespielt wird. Das Ziel des Spiels besteht darin, die Astronautin zum versteckten Kristall zu führen. Für die Steuerung der Astronautin wird die Maus verwendet. Es wurde eine Variable «Temperatur» programmiert, die anzeigt, wie weit die Astronautin vom Kristall entfernt ist. Dieser Abstand ergibt sich aus einer Skalierung der mittels des Blocks «Entfernung von» gemessenen Entfernung zwischen Astronautin und Kristall. Die Skalierung und Zuordnung der Temperaturhinweise sind in Abbildung 18 zu erkennen. Sobald die Entfernung weniger als 10 Pixel beträgt, wird ein Klang gespielt, der Kristall erscheint kurz, der Punktestand erhöht sich auf 1 und der Kristall versteckt sich wieder an einem neuen Ort. Das Spiel endet nach 90 Sekunden mit dem Hinweis, wie viele Punkte gesammelt wurden.

Abbildung 18

Auszug aus dem Code des Spiels <Temperatur>.

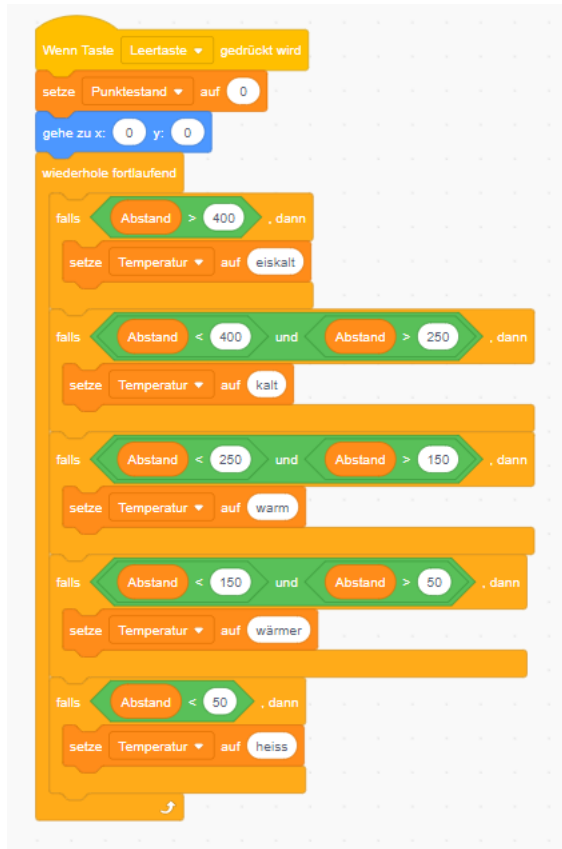
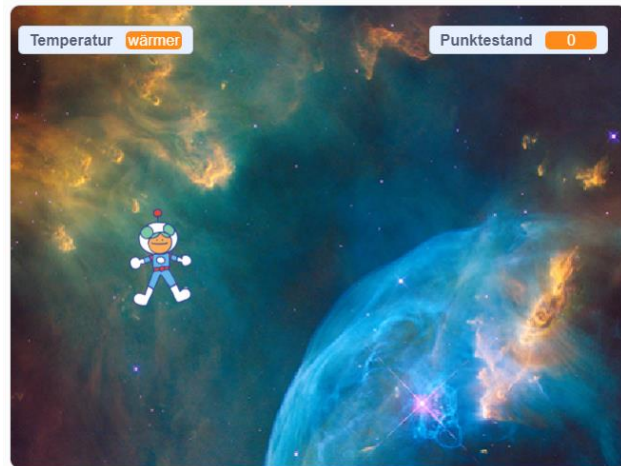


Abbildung 19

Hintergrund des Spiels <Temperatur>



6.4.3 Probeinterview

Im April 2024 haben zwei Probeinterviews stattgefunden. Die Interviews wurden mit zwei Kindern aus der eigenen Klasse durchgeführt. Diese Probeinterviews offenbarten vier Schwachstellen im Aufbau des Interviews, die entsprechend angepasst wurden.

1. Der erste Themenblock wurde von den beiden Kindern deutlich kürzer beantwortet als erwartet. Die Kinder gaben selbst an, dass sie den Anfang als anspruchsvoll empfunden hätten. Aus diesem Grund wurden zwei weitere leichte Einstiegsfragen formuliert. Gemäss Wehr (2014) können solche einfachen Erzählanreize den Kindern helfen, sich in das noch unbekannte Setting einzufinden.
2. Bei der Think-Aloud-Phase haben beide Kinder ausführlich ihr Handeln kommentiert. Die beiden Kinder kannten die interviewende Person jedoch gut. Damit sichergestellt werden kann, dass gewisse Inhalte in den Beschreibungen vorkommen, und um eine Unterstützung für Kinder zu schaffen, die in dieser Phase eher wenig erzählen, wurden zu jedem

Programm abschliessende Fragen hinzugefügt. Diese inkludieren spezifische Fragen zu den sichtbaren Variablen.

3. Bei den Scratchprogrammen wurde eine schriftliche Anleitung mithilfe einer Sprechblase programmiert. Die programmierte Zeitspanne war für die Kinder jedoch zu kurz. Es wurde deshalb bei allen Programmen eine Startfolie mit Anleitung programmiert, die über das Betätigen der Leertaste weggeklickt werden konnte. Damit wurde sichergestellt, dass die Lernenden erst mit dem Programm starten, wenn keine Fragen mehr offen sind.
4. Der letzte Teil des Leitfadeninterviews war für die beiden Kinder am anspruchsvollsten. Zusätzlich muss beachtet werden, dass je nach Formulierung der Frage bereits gewisse Hinweise über das Programmierkonzept «Variablen» abgegeben werden. In Absprache mit der Betreuerin dieser Arbeit wurde entschieden, den Lernenden spezifische Hinweise zu geben. Nach jedem Hinweis werden die Lernenden aufgefordert, den Begriff «Variable» erneut zu erläutern. Allen Teilnehmenden werden der Reihe nach alle Hinweisstufen gegeben (vgl. Tabelle 15).

Tabelle 15

Auszug aus dem Interviewleitfaden über das Programmierkonzept «Variable»

Erzählaufforderung	Du hast mir bereits viel Spannendes erzählt. In unserem Gespräch kamen verschiedene Variablen vor. Hast du eine Idee, was eine Variable sein könnte?
Nachfrage (falls Antwort «nein» oder «keine Idee»)	Es macht nichts, wenn du nicht weisst, was das ist. Du kannst mir deine Vermutung erzählen. Was könnte das sein?
Hinweisniveau 1	Die Variablen kamen auch in den Programmen und Spielen vor, die wir angeschaut haben.
Hinweisniveau 2	Was haben Zeit, die Temperatur und die Geschwindigkeit in den Programmen gemeinsam?
Hinweisniveau 3	Welches Spiel ist dir gut in Erinnerung geblieben? In diesem Spiel waren X und X die Variablen. Was haben sie in diesem Programm gemacht?

6.5 Prozedur

Im April 2024 wurden vier Lehrpersonen angeschrieben. Zwei Lehrpersonen haben einer Teilnahme zugestimmt und den Informationsbrief mit der Einverständniserklärung (Anhang 10.3.1) der ganzen Klasse ausgehändigt. Aus der ersten Klasse haben neun von sechzehn Eltern ihr Einverständnis gegeben. In der zweiten Klasse haben sich fünf von fünfzehn Kinder angemeldet. Laut der Lehrperson ist diese kleine Rücklaufquote darauf zurückzuführen, dass einige Lernende kein Interesse an Scratch haben. Kinder mit speziellen Förderbedürfnissen wie angepasste Lernziele oder Deutsch als Zweitsprache wurden von der Klassenlehrperson vorgängig aussortiert, da die Interviewsituation als eher anspruchsvoll eingeschätzt wurde. Aus der ersten Klasse wurden randomisiert fünf der neun Kinder ausgewählt; aus der zweiten Klasse haben alle fünf Kinder teilgenommen.

Die Auskunftsbereitschaft von Kinder kann durch vertraute Räumlichkeiten mit möglichst wenig Ablenkungen gefördert werden (Vogl, 2021). Die Interviews wurden deshalb im Mai 2024 während der Unterrichtszeit in einem Gruppenraum durchgeführt. Alle Interviews einer Klasse haben an einem Unterrichtsmorgen in der Schule stattgefunden. Die interviewende Person hat sich zu Unterrichtsbeginn der Klasse vorgestellt, den Interviewablauf kurz beschrieben und sich für die Teilnahme der Kinder bedankt.

Vom Interview wurde eine Audioaufnahme gemacht. Zu Beginn wurden dem jeweiligen Kind das Ziel und der Ablauf des Interviews noch einmal erklärt. Der Interviewleitfaden (Anhang 10.3.2) lag in gedruckter Form vor. Das Interview wurde auf Hochdeutsch geführt. Die interviewten Kinder haben die Scratchspiele am Laptop selbst bedient, wofür sie meistens die Maus verwendet haben. Ein Kind hat die Touchfunktion des Bildschirms genutzt. Die Kinder durften die Anleitungen der Spiele selbst lesen oder sie wurden von der interviewenden Person vorgelesen. Während des Interviews wurden Notizen zu Auffälligkeiten während der Bedienung der Spiele aufgezeichnet. Im Durchschnitt dauerten die Interviews 37 Minuten mit einer Range von 30–53 Minuten. Diese Informationen können der Beschreibung der Stichprobe in Anhang 10.3.3 entnommen werden.

6.6 Ablauf der Datenauswertung

Die Audiodateien der Interviews wurden unmittelbar nach der Durchführung mithilfe der Transkriptionssoftware MAXQDA Transcription transkribiert. Es wurde die Sprache «Deutsch» ausgewählt und das Vokabular «Variable», «Scratch» und «Uhrzeit³⁰» eingefügt. Das erstellte Transkript wurde anhand der Audiodatei überarbeitet. Bei der Transkription wurden die

³⁰ Im Probeinterview wurde dieses Wort immer klein geschrieben, weshalb es ins Vokabular aufgenommen wurde.

Transkriptionsregeln des inhaltlich-semanticen Transkriptionssystems nach Dresing und Pehl (2018) geachtet. Auf eine Angabe von Pausen wurde verzichtet, da während der Verwendung der Spiele teilweise lange Pausen entstanden sind. Die Transkripte der Interviews können dem Anhang 10.3.5-10.3.14 entnommen werden.

Der anschliessende Ablauf der Datenauswertung folgte der inhaltlich strukturierenden qualitativen Inhaltsanalyse gemäss Kuckartz und Rädiker (2022). Dazu wurden in einem ersten Schritt kurze stichwortartige Fallzusammenfassungen angelegt, die besondere Auffälligkeiten beinhalten. Dazu wurden auch die Notizen aus den Interviews herangezogen. In einem zweiten Schritt wurden die Hauptkategorien entwickelt, die sich aus dem Aufbau des Leitfadens und den Forschungsfragen ergeben haben. Die Hauptkategorien lauten «Programmiererfahrung», «Beschreibung Zeit», «Beschreibung Temperatur», «Beschreibung Geschwindigkeit» sowie «Definition Variable mit den unterschiedlichen Hinweisstufen». In einem ersten Kodierungsprozess wurde gemäss dieser Hauptkategorien kodiert. Diese kodierten Textstellen wurden nun studiert und daraus passende Subkategorien entwickelt. Diese induktive Entwicklung der Subkategorien wurde für jede Hauptkategorie vorgenommen. Zur Kategorie wurde eine Definition verfasst und auf ein passendes Ankerbeispiel aus einem der Interviews verwiesen. Bei der Beschriftung der Ausprägung wurde darauf geachtet, den möglichst genauen Wortlaut der Lernenden zu verwenden. In einem zweiten Kodierungsprozess wurden nun die Transkripte unter Berücksichtigung der neuen Subkategorien kodiert. Der finale Kodierleitfaden kann dem Anhang 10.3.4 entnommen werden.

6.7 Ergebnisse

Im Folgenden werden FF3 «Welche Übereinstimmungen lassen sich zwischen dem Programmierkonzept «Variablen» und den Beschreibungen der Alltagsvariablen Zeit, Temperatur und Geschwindigkeit von Lernenden im Alter von 10-12 Jahren feststellen?» und FF4 «Wie beschreiben Kinder im Alter von 10–12 Jahren das Programmierkonzept «Variablen», nachdem sie verschiedene Scratchprogramme mit Variablen angeschaut haben?» anhand der Ergebnisse aus den Interviews mit den Kindern beantwortet. Die Ergebnisse werden vornehmlich in Form eines Fliesstextes zusammengefasst. Sofern es das Verständnis fördert, wird ein Auszug aus den Interviews angefügt. Zur Beantwortung von FF3 werden die Beschreibungen der Kinder zu den Alltagsvariablen «Zeit», «Temperatur» und «Geschwindigkeit» dargestellt. Diese Beschreibungen haben die Lernenden formuliert, bevor sie eines der Scratchprogramme bedient haben. Zur Beantwortung von FF4 werden die Beschreibungen der Kinder zum Programmierkonzept «Variablen» dargestellt, die nach der Bedienung aller Scratchprogramme formuliert wurden.

6.7.1 Beschreibung der Lernenden über Zeit

Wie in Kapitel 6.4.1 beschrieben, werden die Lernenden im ersten Teil jedes Themenblocks dazu aufgefordert, zu erklären, was sie darüber wissen. In diesem Kapitel geht es darum, wie die Lernenden das Konzept «Zeit» beschrieben haben, bevor sie die passenden Scratchprojekte gespielt haben. Die Antworten der Lernenden konnten verschiedenen Kategorien zugeordnet werden, die im Folgenden aufgeführt werden sollen.

Uhrzeit

Nahezu alle Kinder (Interview 2, 3, 4, 5, 6, 7, 8, 10) haben in ihrer Beschreibung darauf hingewiesen, dass die Zeit mit der Uhrzeit angegeben werden kann. Dabei haben die Lernenden teilweise erwähnt, dass die Uhrzeit auf einer Uhr oder auch auf dem Handy abgelesen werden und so in Erfahrung gebracht werden kann, wie spät es ist. Die Kinder haben hierbei oft erwähnt, dass sie in der Schule gelernt hätten, die Uhrzeit zu lesen.

Ein Kind (Interview 5) hat angegeben, dass auch Zukunft und Vergangenheit für ihn zum Thema «Zeit» gehören würden.

Messung der Zeit

Zwei Kinder (Interview 2,4) haben erwähnt, dass die Zeit mithilfe einer Stoppuhr gemessen werden kann, beispielsweise im Rahmen eines Sportwettkampfs. Mehrere Lernende haben unterschiedliche Zeiteinheiten aufgezählt, die ihnen bekannt sind. Dabei wurden dabei Millisekunden, Sekunden, Minuten, Stunden, Tage, Wochen und Monate genannt.

Wahrnehmung der Zeit

Bei der Subfrage, was die Kinder interessant an der Zeit finden, haben einige Kinder (Interview 1, 4, 6, 9) erzählt, dass sie es interessant finden würden, dass die Zeit nicht immer gleich schnell vergehe. Damit gingen Beschreibungen von Situationen einher, in denen die Zeit besonders schnell vergangen sei oder sich umgekehrt langsam angefühlt habe. Ein Kind (Interview 9) hat beim Themenbereich «Geschwindigkeit» ausführlich beschrieben, wie die Zeit bei gewissen Aktivitäten schneller oder auch langsamer vergehen kann.

Zwei Kinder (Interview 3, 7) haben auch erwähnt, dass Zeit für sie stressig sei, weil sie manchmal Aufgaben auf das Ende der Lektion in der Schule fertig machen muss.

Beschreibung eines Zeitkontinuums

Einige Lernende (Interview 1, 2, 3, 5, 7, 10) haben beschrieben, dass die Zeit schon immer da gewesen sei und unaufhaltsam weiterlaufe. Eines der Kinder gab folgende Beschreibung an:

Man ist also man kann also eigentlich würde ich mal so sagen, alles besteht eigentlich auch aus Zeit. Also das ganze Leben besteht aus Zeit. Man ist immer in einer Zeit halt, die kann man nicht einfach so stoppen, die läuft einfach immer weiter. Und das finde ich ein bisschen speziell, ja. (Interview 5, Pos. 118)

Einfluss der Zeit auf den Alltag

Einige Kinder (Interview 1, 2, 4, 5, 6, 7, 8 und 9) haben beschrieben, dass die Zeit einen Einfluss darauf habe, was sie machen würden. Beispielsweise müsse man zu einer bestimmten Zeit in die Schule gehen oder aufstehen. Den Aussagen der Kinder zufolge strukturiert die Zeit den Ablauf des Tages, damit klar wird, was wann gemacht werden muss. In einer Subfrage wurde explizit nach dem Einfluss der Zeit auf den eigenen Alltag gefragt. Einige Kinder haben daraufhin detailliert ihren zeitlichen Alltag skizziert.

Halt in der Schule, also habe ich halt fünf Stunden Schule. Das weiss ich noch von 8.00 bis 11:20 und dann noch am Mittag. Dann habe ich noch halb um 5:30 bis 7 habe ich dann noch Fussballtraining. Also das ist auch noch mit Zeit. Dann habe ich jeden Samstag noch ein Match, der auch eine halbe Stunde, also eineinhalb Stunden geht und ja, das beeinflusst halt mein Tag oder so, ja. (Interview 6, Pos. 122)

6.7.2 Beschreibung der Lernenden über Temperatur

Im Folgenden werden die Aussagen der Kinder zum Konzept «Temperatur» präsentiert, die vor dem Spielen der Scratchprojekte getätigt wurden. Die Auswertung der Aussagen konnte in Kategorien zusammengefasst werden, die im Folgenden präsentiert werden sollen.

Temperaturskalen

Drei Kinder (Interview 7, 8, 10) haben beschrieben, dass Temperatur in Grad Celsius angegeben werden kann. Zwei dieser drei Kinder (Interview 7, 10) sind darauf eingegangen, dass auch Fahrenheit als Masseinheit verwendet werden kann.

Messung der Temperatur

Einige Kinder (Interview 1, 3, 4, 7, 9) sind in ihren Beschreibungen darauf eingegangen, dass Temperatur gemessen werden kann. Teilweise haben sie dazu spezifische Orte für die Temperaturmessung genannt (z. B. den Pool oder die Lufttemperatur). Wenige Kinder (Interview 1, 3, 9) haben erwähnt, dass die Temperatur mit einem Thermometer gemessen werden kann.

Zustand der Temperatur

Alle Lernenden haben Aussagen über die Zustände der Temperatur gemacht. Dabei haben sie meist erwähnt, dass Temperatur warm oder kalt sein kann. Ein Kind (Interview 10) hat diese Ausprägungen mit den Farben Rot (heiss) und Blau (kalt) in Verbindung gebracht.

Wetter

Wenige Kinder (Interview 3, 5, 8) haben Phänomene wie Wind, Regen, Schnee, Eis oder explizit das Wetter genannt.

Veränderlichkeit von Temperatur

In den Interviews 2, 3 und 7 haben die Lernenden erwähnt, dass sie es als spannend empfinden würden, dass die Temperatur sich verändern kann. Ein Kind hat dies folgendermassen beschrieben: «Ich finde es spannend, dass man das so erkennen kann, wie kalt oder warm es ist. [...] Und wie krass sich das verändern kann.» (Interview 3, Pos. 78-80)

Einfluss der Temperatur auf ...

Bei der Subfrage nach dem Einfluss der Temperatur auf den Alltag haben die Kinder unterschiedliche Gebiete aufgezählt, auf die Temperatur einen Einfluss haben kann. Die Aussagen der Kinder können folgenden Kategorien zugeordnet werden:

... den Körper

Einige Kinder (Interview 5, 6, 7, 8, 9) haben beschrieben, dass die Temperatur einen Einfluss auf den eigenen Körper haben kann. Ein Kind gab hierzu Folgendes an:

Äh, also, wenn es kalt ist, dann frieren wir halt. Also, es hat eigentlich einen Einfluss auf unseren Körper. Wenn es kalt ist, frieren wir. Und wenn es zu heiss ist, schwitzen wir. Und ja, unser Körper will, glaube ich, immer eigentlich auf der gleichen Temperatur bleiben. Deshalb macht er das, um die Temperatur auszugleichen eigentlich. (Interview 7, Pos. 138)

Zwei Kinder (Interview 8, 9) haben beschrieben, dass der Mensch Fieber haben kann. Ein Kind (Interview 5) hat erwähnt, dass es möglich ist, bei zu tiefer Temperatur zu erfrieren.

... Erde und Weltraum

Einige Lernende (Interview 1, 4) haben ausgesagt, dass die Temperatur einen Einfluss auf die Erde haben könne. Laut den Lernenden ist die Temperatur der Grund dafür, dass es auf der Welt nicht überall gleich aussieht. Ein Kind (Interview 5) hat beschrieben, dass solche Einflüsse auch im Weltall gelten würden.

... die Aktivitäten

Die Temperatur hat laut einigen der Kinder (Interview 1, 3, 5, 10) einen Einfluss auf Aktivitäten, denen sie nachgehen. Dabei wird unterschieden zwischen Aktivitäten, die bei warmen Temperaturen passend erscheinen, wie Glace essen, rausgehen, Fussball spielen oder auch Schwimmen und Aktivitäten, denen bei kalten Temperaturen nachgegangen wird, wie Ski fahren.

... die Kleidung

Einige Lernende (Interview 1, 2, 3, 4, 5, 6, 9, 10) haben den Einfluss der Temperatur auf ihre Kleidung beschrieben. Dabei werden im Kontext wärmerer Temperaturen weniger Jacken oder kürzere Hosen erwähnt. Sind die Temperaturen kalt, ziehen die Kinder wärmere Kleider an. Wenige Kinder haben die Aussagen zu den Kleidern an die Jahreszeiten Winter und Sommer gekoppelt.

6.7.3 Beschreibung der Lernenden über Geschwindigkeit

Wie bei den beiden anderen Bereichen wurden die Lernenden gebeten, zu beschreiben, was sie über Geschwindigkeit wissen. Die Antworten der Lernenden vor dem Spielen der Scratchprogramme konnten in Kategorien zusammengefasst werden, die im Folgenden thematisiert werden.

Masseinheiten von Geschwindigkeit

Zwei Kinder (Interview 6, 7) haben in ihren Beschreibungen von Kilometern pro Stunde gesprochen. Eines dieser Kinder (Interview 7) hat erklärt, dass Geschwindigkeit mit dieser Angabe gemessen wird. Das andere Kind (Interview 6) hat angegeben, dass auf Englisch von «Miles per hour» gesprochen wird.

Zustand der Geschwindigkeit und andere Bezeichnungen

Der Zustand der Geschwindigkeit wurde von den Lernenden mit unterschiedlichen Ausdrücken beschrieben. Oft wurde angegeben, dass etwas schnell oder langsam sein kann (Interview 1, 3, 5, 8). Ein Kind (Interview 6) hat Geschwindigkeit mit Schnelligkeit verglichen. Ein weiteres Kind (Interview 5) hat die Geschwindigkeit auch «Tempo» genannt.

Geschwindigkeit bei unterschiedlichen Fortbewegungsmitteln

Die Geschwindigkeit wurde von den Lernenden meist in Verbindung mit einer Fortbewegungsart beschrieben. Dabei wurde die Geschwindigkeit beim Laufen beziehungsweise Rennen (Interview 1, 3, 4, 5, 7, 8, 10), beim Autofahren (Interview 1, 2, 3, 7, 8), beim Skifahren (Interview 2), beim Fliegen (Interview 8) oder bei einem Scooter (Interview 8) genannt.

Weltrekorde mit Geschwindigkeit

In den Interviews wurden zwei unterschiedliche Weltrekorde erwähnt, die mit Geschwindigkeit im Zusammenhang stehen. Ein Kind (Interview 2) hat die höchste gemessene Geschwindigkeit beim Skifahren erwähnt, ein anderes Kind (Interview 6) den schnellsten Mann der Welt. Ein Kind (Interview 4) hat angegeben, dass manche Tiere schnell beschleunigen können.

Einfluss auf die Geschwindigkeit

Die Kinder wurden gefragt, welchen Einfluss die Geschwindigkeit auf ihren Alltag hat. Die Antworten der Kinder sind bei diesem Thema deutlich kürzer ausgefallen als die Antworten auf die gleiche Frage zu den Themenbereichen «Temperatur» und «Zeit». In ihren Beschreibungen haben die Kinder eher thematisiert, wie bestimmte Aspekte einen Einfluss auf die Geschwindigkeit haben.

Beispielsweise haben Kinder erwähnt, dass sie schneller laufen, um nicht zu spät zu kommen (Interview 1, 6).

6.7.4 Beschreibung der Lernenden über Variablen

Im letzten Teil des Interviews, nachdem alle Scratchprogramme angeschaut wurden, wurden die Kinder aufgefordert, zu erklären, was für sie eine Variable ist. Dabei wurden den Kindern in einer abgestimmten Reihenfolge Hinweise gegeben (vgl. Tabelle 15). Im Folgenden werden die Antworten der Kinder in zusammengefassten Kategorien präsentiert. Sofern es als relevant erachtet wurde, ist zudem angegeben, auf welcher Hinweisstufe die Antworten gegeben wurden.

Variablen als «verschiedene Sachen»

Drei Kinder (Interview 1, 2, 5) haben angegeben, dass Variablen unterschiedliche Sachen sein könnten. Ein Kind (Interview 1) hat dabei eine Verbindung zum Wort «Varianten» hergestellt. Die drei Kinder haben diese Vermutung bei der ersten Erzählaufforderung getroffen.

Variablen sind Zahlen

Drei Kinder (Interview 2, 3, 5) haben beschrieben, dass Variablen etwas mit Zahlen zu tun hätten. Ein Kind (Interview 5) hat erklärt, dass bei diesen Zahlen etwas eingegeben werden müsse. Ein weiteres Kind (Interview 2) hat auf jeder Hinweisstufe diese Aussage getroffen. Kein Kind hat jedoch beschrieben, dass Variablen nur Zahlen sind.

Variablen sind veränderbar

Vier Kinder (Interview 6, 7, 9, 10) beschrieben Variablen als etwas Veränderliches. Beispielsweise gab ein Kind (Interview 9) Folgendes an: «Die Variablen sind Sachen, die etwas

eingeben, angeben oder vielleicht auch die du verändern kannst. Hier kannst du zum Beispiel die Geschwindigkeit verändern.» (Interview 9, Pos. 150)

Temperatur, Zeit und Geschwindigkeit sind Variablen

Mehrere Kinder (Interview 3, 4, 5, 7, 8, 10) haben mindestens eine der drei Variablen ‹Temperatur›, ‹Zeit› und ‹Geschwindigkeit› als Variablen erkannt und diese als solche benannt. Zwei Kinder (Interview 4, 10) haben Temperatur, Zeit und Geschwindigkeit als Variablen des eigenen Lebens beschrieben.

Variablen können Sachen speichern

Zwei Kinder (Interview 7, 10) gaben an, dass mit Variablen Sachen abgespeichert bzw. gemerkt werden könnten. Ein Kind hat es folgendermassen formuliert: «Variablen sind um etwas zu merken eigentlich auch für mich. Dann setzt das zum Beispiel auf vier und dann merkt es sich das und auf Punktestand halt.» (Interview 7, Pos. 184)

Variablen beschreiben etwas

Drei Kinder (Interview 2, 8, 9) gaben an, dass Variablen für sie etwas beschreiben oder auch anzeigen würden. Ein Kind erwähnte hierzu folgendes Beispiel: «Sie beschreiben etwas also man kann mit ihnen etwas beschreiben. Also ich würde sagen, das Auto fährt 30 km/h und ist 20 Grad warm.» (Interview 2, Pos. 239)

Variablen können etwas auslösen

Zwei Kinder (Interview 7, 9) beschrieben, dass Variablen in einem Programm einen Einfluss hätten. Das Kind in Interview 9 erklärte, dass Variablen eine Art Steuerung seien. Das Kind in Interview 7 beschrieb den Einfluss wie folgt: «Und dann, falls eine bestimmte Zahl ist, löst das auch etwas aus oder so.» (Interview 7, Pos. 188)

Variablen als vielseitiges Programmierkonzept

Ein Kind (Interview 10) ordnete Variablen als drittwichtigsten Block in Scratch ein. Das Kind verwendete diese in schwierigeren Spielen beispielsweise für den Punktestand. Weiter führte das Kind aus, dass mit Variablen auch verschiedene Levels in einem Spiel programmiert werden könnten. Ein weiteres Kind (Interview 7) gab an, dass mit Variablen andere Sachen programmiert werden könnten als mit den anderen Blöcken.

Variablen werden benannt

Ein Kind (Interview 7) sagte aus, dass Variablen einen Namen hätten und dadurch bekannt sei, was darin gespeichert wird.

Eine Variable ist eine Box

Ein Kind beschrieb eine Variable als einen «Kasten oder eine Box, in dem etwas steht, was man also was, was das ist.» (Interview 8, Pos. 142)

Variablen sind nicht selbst bestimmbar

Ein Kind (Interview 1) gab an, Variablen seien etwas, das nicht selbst bestimmt werden könne.

6.7.5 Zusätzliche Befunde

Neben den für FF3 und FF4 relevanten Ergebnissen sind zusätzliche Befunde ermittelt worden, über die in diesem Kapitel berichtet werden soll.

Kenntnis des Begriffs «Variable»

Die interviewende Person hat erst im letzten Teil des Interviews das Wort «Variable» verwendet. Drei Kinder (Interview 6, 7 und 10) haben den Begriff bereits von sich aus während des Bedienens der Scratchprogramme verwendet. Teils wurde dabei die Variable benannt: «Eine Variable mit einem Punktestand.» (Interview 7, Pos. 170). In einzelnen Fällen wurde genauer erklärt, wie der Wert dieser Variable zustande kommt: «Das ist, glaube ich, auch eine Variable wieder. Und die wird nach jeder Sekunde eigentlich soll sie um einen erhöhen, glaube ich.» (Interview 7, Pos. 68)

Ein Kind (Interview 9) hat das Wort «Variable» nicht benutzt, aber von den orangen Blöcken gesprochen, die verwendet werden können, um den Punktestand zu programmieren.

Ein weiteres Kind (Interview 8) hat angegeben, das Wort «Variable» bereits aus dem Themenbereich «Zahl und Variable» im Mathematikunterricht zu kennen.

Alle Kinder, die den Begriff «Variable» bereits gehört oder selbst verwendet haben, kommen aus der gleichen Klasse.

Think-Aloud

Die Phase des Bedienens der Scratchprogramme wurde in Bezug auf FF3 und FF4 nicht kodiert. Sie diente in erster Linie der unbewussten Konfrontation der Lernenden mit Variablen. Die Lernenden wurden jeweils gebeten zu erklären, wie das Programm funktioniert. Einige Lernende haben dabei beschrieben, was sie beobachten konnten. In Interview 1 erklärte das Kind das Spiel folgendermassen: «Der Timer startet von null und dann fährt es einfach hoch. Der Frosch geht durch das Labyrinth. Wenn man an den Rand ankommt, wird der Spielstand zurückgesetzt. Und man kann dann wieder loslegen.» (Interview 1, Pos. 173) Andere Kinder beschrieben, wie dieses Spiel programmiert wurde:

Man nimmt ein Bühnenbild und man nimmt die Katze. Man programmiert die Katze so, dass man 15er Schritte nach rechts und nach links laufen kann. Dann muss man die Katze auch programmieren. Falls der Rand, also falls der Rand berührt wird, wechselt das Bühnenbild und dann wechselt das Bühnenbild auf zum Beispiel die kalte Landschaft. Und dann kann man auch programmieren, dass dann, falls das Bühnenbild wechselt, dass sie das und das sagen sollte, also kann. Und oben rechts hat man das auch aus Figuren genommen, die Brille und so. Dort hat man dann programmiert, falls wird das angeklickt, senden Nachricht an Katze eins, also die grosse Katze in der Mitte, wo man programmieren kann. Und dann bei der grossen Katze, falls die Katze Nachricht empfängt, wechsele Kostüm zu Kappe und Schal und dann geht es halt immer so weiter und weiter und links oben ist jedes Mal die Temperatur. Das kann man auch programmieren. Also falls das Bühnenbild auf zum Beispiel Mexiko wechselt: stelle Temperatur auf 30 Grad. Also ja, das kann man so programmieren. (Interview 6, Pos. 60)

In dieser Phase wurden die Lernenden auch darum gebeten, die sichtbare Variable zu beschreiben³¹. Hierbei nannten die Kinder meist den Namen der Variable. Manchmal teilten sie mit, wie der Wert dieser Variable zustande kommt: «Mein Punktestand oder der Punktestand. Und da werden die Punkte oder die Kristalle, die ich gesammelt habe, wird dann gezählt dort.» (Interviews 4, Pos. 158) Insbesondere bezüglich der Variable «Punktestand», die in zwei Spielen vorkam, konnten mehrere Lernende erklären, was der Wert in der Variable bedeutet.

6.8 Zwischendiskussion und Fazit

In diesem Kapitel werden FF3 und FF4 beantwortet und zusätzliche Befunde präsentiert. Es wird aufgezeigt, welche Implikationen die Ergebnisse aus U3 für die vorliegende Arbeit haben.

6.8.1 Beantwortung von FF3

Um FF3 «Welche Übereinstimmungen lassen sich zwischen dem Programmierkonzept «Variablen» und den Beschreibungen der Alltagsvariablen Zeit, Temperatur und Geschwindigkeit von Lernenden im Alter von 10-12 Jahren feststellen?» zu beantworten, wurden Lernende der 6. Klasse in einem Leitfadeninterview nach ihrer Vorstellung zu der benannten Alltagsvariablen

³¹ Die interviewende Person hat dafür auf die Variable gezeigt und gefragt: Was ist das?

gefragt. Diese Beschreibungen wurden kategorisiert und dargestellt. Die Kategorien werden im Folgenden mit dem Programmierkonzept «Variablen» verglichen.

Die grösste Übereinstimmung besteht zwischen dem Programmierkonzept «Variablen» und den Beschreibungen der Lernenden zur Veränderlichkeit von Temperatur. Einigen Lernenden in diesem Alter scheint die Veränderlichkeit von Temperatur aufzufallen. Auch Variablen können ihren Wert, je nach Rolle im Programm, verändern (Fischer & Hofer, 2011). Bei Zeit und Geschwindigkeit sind keine solche direkten Beschreibungen über Veränderlichkeit zu finden.

Die Beschreibungen der Kinder lassen weitere mögliche Übereinstimmungen erahnen, die jedoch nicht die gleiche Deutlichkeit aufweisen.

Bei der Geschwindigkeit und der Temperatur haben die Lernenden jeweils unterschiedliche Zustände beschrieben. Diese Zustände weisen meist einen bipolaren Charakter auf (heiss und kalt, schnell und langsam). Hier kann ein Vergleich zu den Wahrheitswerten (wahr und falsch, auch «Booleans» genannt) gezogen werden, die einem möglichen Datentyp von Variablen entsprechen (Grover & Basu, 2017). Wenn der Fokus jedoch vermehrt auf dem Wechsel zwischen den unterschiedlichen Zuständen liegt, kann hier auch eine Verbindung zur zuvor beschriebenen Veränderlichkeit der Werte von Variablen hergestellt werden.

Wenige Kinder haben sich über Rekorde geäussert, die mit Geschwindigkeit in Verbindung gebracht werden können. Als Rekord wird eine Leistung bezeichnet, die die bisher beste darstellt. Wie in Kapitel 2.3.4 beschrieben, können Variablen in Programmen verschiedenen Rollen zugeordnet werden (Sajaniemi et al., 2006). Eine solche Weltrekordvariable würde der Variable «Most-wanted-holder» entsprechen.

Vor allem bei der Temperatur, aber auch im Hinblick auf Zeit und Geschwindigkeit gaben die Kinder unterschiedliche Einflüsse auf ihren Alltag an. Die Formulierungen der Kinder zeigen, dass diese in ihrem Alltag Abhängigkeiten in Bezug auf die Alltagsvariablen erkennen. Im Bereich der Informatik wird in diesem Fall von «Bedingungen» gesprochen (Brennan & Resnick, 2012). Bei diesen kommen oft Variablen zum Einsatz. In U2 bestätigte eine Expertin den engen Bezug zwischen den beiden Programmierkonzepten «Bedingung» und «Variablen». Es scheint für Kinder verständlich zu sein, dass gewisse Aspekte beispielsweise einen Einfluss auf die Kleiderwahl oder die Freizeitgestaltung haben können. Das könnte bedeuten, dass sich die drei Alltagsvariablen für die Veranschaulichung von Bedingungen eignen könnten.

Bei der Temperatur haben einige Kinder darüber gesprochen, dass diese mithilfe eines Thermometers gemessen werden kann. Wenige Kinder haben im Kontext der Zeit den Messvorgang mit einer Stoppuhr erwähnt. Wenn etwas gemessen wird, wird ein Wert abgelesen. Wenngleich die Lernenden dies nicht direkt angegeben haben, findet bei einer solchen Messung stets eine Zuweisung des gemessenen Werts zu der Alltagsvariable, in diesem Fall der

Temperatur, statt. Es könnte also sinnvoll sein, den Vorgang des Messens zu nutzen, um Lernenden zu erklären, dass der Wert, der anschliessend abgelesen wird, der Variable ‹Temperatur› zugewiesen wird. Anders formuliert: An einem Thermometer können Lernende den aktuellen Wert der Variable ‹Temperatur› ablesen.

Bei der Beschreibung der Zeit haben einige Lernende darüber gesprochen, dass es Zeit schon immer gegeben habe und dass diese unaufhaltsam weiterlaufe. Aus der Sicht einiger Kinder kann die Zeit also nicht beeinflusst werden. Variablen hingegen werden von programmierenden Personen durch das Erstellen eines Programms initialisiert und verändert. E1 aus U2 sieht genau in diesem Punkt eine mögliche Schwierigkeit der Alltagsanalogien: Über die Alltagsvariablen besteht nicht die gleiche Kontrolle wie eine programmierende Person sie über die Variablen ausüben kann.

Abschliessend deuten die Ergebnisse aus U3 bezogen auf FF3 darauf hin, dass die Alltagsvariable ‹Temperatur› die grösste Übereinstimmung mit dem Programmierkonzept ‹Variablen› aufweist: die Veränderlichkeit von Werten.

6.8.2 Beantwortung von FF4

Um FF4 ‹Wie beschreiben Kinder im Alter von 10–12 Jahren das Programmierkonzept ‹Variablen›, nachdem sie verschiedene Scratchprogramme mit Variablen angeschaut haben?› zu beantworten, wurden die Lernenden in einem Leitfadeninterview auf unterschiedlichen Hinweisstufen darum gebeten, zu erklären, was für sie eine Variable ist. Die Lernenden in U3 beschrieben Variablen den folgenden Kategorien: ‹Variablen als verschiedene Sachen›, ‹Variablen sind Zahlen›, ‹Variablen sind veränderbar›, ‹Temperatur, Zeit und Geschwindigkeit sind Variablen›, ‹Variablen können Sachen speichern›, ‹Variablen beschreiben etwas›, ‹Variablen können etwas auslösen›, ‹Variablen als vielseitiges Programmierkonzept›, ‹Variablen werden benannt›, ‹Eine Variable ist eine Box› und ‹Variablen sind nicht selbst bestimmbar›.

Einige dieser Beschreibungen der Lernenden können als intuitive Vorstellungen von Variablen betrachtet werden, die einfachen Verallgemeinerungen entsprechen, die sich im Alltag oder über soziale Interaktion entwickeln (Babari et al., 2023). Dazu zählen auch ungenaue Beschreibungen oder solche, in denen Fachbegriffe fehlen. Ein Beispiel hierfür sind die Kategorien ‹Variablen als verschiedene Sachen› oder ‹Variablen beschreiben etwas›. Die Formulierungen der Lernenden in dieser Kategorie sind allgemein und simpel gehalten. Intuitive Vorstellungen können auch teilweise korrekt sein – so z. B. die Beschreibung ‹Variablen sind Zahlen›. Sie ist nur teilweise korrekt, weil Variablen auch Werte mit anderen Datentypen speichern können.

Andere Beschreibungen der Lernenden weisen Überschneidungen mit bekannten Eigenschaften des Programmierkonzepts *Variable* auf, die in Tabelle 3 beschrieben sind. Dazu zählen die Aussagen «Variablen sind veränderbar» (Veränderbarkeit) und «Variablen werden benannt» (Benennung). Die Beschreibungen in der Kategorie «Variablen als vielseitiges Programmierkonzept» offenbaren ein Verständnis über den Einsatz von Variablen. Die drei Kinder, die Aussagen zu diesen drei Kategorien gemacht haben, haben das Wort *Variable* von sich aus während des Interviews verwendet. Diese Punkte lassen vermuten, dass diese Lernenden bereits eine elaboriertere Vorstellung des Programmierkonzepts *Variablen* haben.

Die Beschreibung «Variable ist eine Box» entspricht der Box-Analogie. Die Beschreibung «Variablen sind nicht selbst bestimmbar» ist in Bezug auf das Programmierkonzept *Variablen* nicht korrekt und könnte deshalb eine FV sein. Sie kann jedoch keiner bisher bekannten FV (Kapitel 2.3) zugeordnet werden.

Die Beschreibungen «Temperatur, Zeit und Geschwindigkeit sind Variablen», «Variablen können Sachen speichern» und «Variablen können etwas auslösen» gehören aufgrund ihres zusammenfassenden Charakters zu den intuitiven Vorstellungen. Es sind jedoch auch Übereinstimmungen mit dem Programmierkonzept *Variable* erkennbar. In dieser allgemeinen Formulierung werden jedoch keine fachlichen Wendungen wie «können Werte speichern» verwendet. Die Aussage «Variablen können etwas auslösen» weist eine Verbindung zum Programmierkonzept *Bedienung* auf. Die Beschreibung «Temperatur, Zeit und Geschwindigkeit sind Variablen» ist grundsätzlich korrekt, stellt jedoch eher einen Fakt dar. Um eine genauere Aussage treffen zu können, müssten die einzelnen Beschreibungen der Lernenden genauer in den Blick genommen und mit den Eigenschaften von Variablen abgeglichen werden.

Zusammenfassend kann festgehalten werden, dass die Lernenden in dieser Untersuchung Variablen nach dem Spielen verschiedener Scratchspiele unterschiedlich elaboriert und meist stark vereinfacht, jedoch selten falsch beschrieben haben.

6.8.3 Implikationen

Die Ergebnisse aus U3 zeigen, dass Lernende nach dem Spielen bestimmter Scratchprogramme dazu in der Lage sind, das ihnen teilweise unbekannte Konzept *Variablen* in eigenen Worten und teilweise stark vereinfacht zu beschreiben. Auch in Ansätzen wie *Use-Modify-Create* (Lytle et al., 2019) wird dem Verwenden und Betrachten von Programmen eine zentrale Rolle im Programmierlernprozess zugeordnet. Die Ergebnisse aus U3 geben Hinweise darauf, dass dies auch beim Verständnis des Programmierkonzepts *Variablen* helfen könnte.

Die Ergebnisse zu FF4 legen die Hypothese nahe, dass die Veränderlichkeit von Variablen jungen Lernenden mithilfe der Alltagsanalogie *Temperatur* erklärt werden könnte. Laut einer

Expertin aus U2 unterscheidet diese Eigenschaft einer Variable die informatische Variable von der mathematischen. Spannend erscheint zudem, dass vereinzelt Lernende im zweiten Teil des Interviews ebendiese Eigenschaft von Variablen beschrieben haben. Ob es sich hierbei um einen Zufall handelt oder ob die vorangegangenen Gespräche einen Einfluss auf die Beschreibungen hatten, kann anhand der Ergebnisse dieser Untersuchung nicht beantwortet werden.

Ein Kind hat in U3 die fehlerhafte Beschreibung formuliert, eine Variable sei etwas, das nicht selbst bestimmt werden könne. Diesen Unterschied zwischen der Alltagsanalogie und dem Programmierkonzept «Variablen» hat auch E1 in U2 beschrieben. Solche Unterschiede zwischen Analogie und Zielkonzept müssen genau betrachtet (Aubusson et al., 2006) und im Unterricht aufgezeigt werden (Treagust et al., 1998).

U1 und U2 haben Hinweise darauf geliefert, dass die Variable «Punktestand» in Aktivitäten zum Programmierkonzept «Variablen» eingesetzt wird. In U3 konnten einige Lernende den Wert dieser Variable nachvollziehen oder sogar erklären. Für den Informatikunterricht mit Variablen bedeutet dies, dass Beispiele mit der Variable «Punktestand» für Lernende in diesem Alter verständlich sein könnten.

Aufgefallen ist auch, dass drei Kinder aus der gleichen Klasse Variablen bereits vor dem Interview gekannt haben. Es handelt sich hierbei um Kinder, die laut der eigenen Beschreibung in der Freizeit häufig programmieren. Die Beschreibungen der Scratchprogramme dieser Kinder haben sich von jenen der anderen unterschieden. Während einige Kinder Programme aus Sicht einer benutzenden Person beschrieben haben, erklärten die drei erwähnten Kinder eher, wie das Spiel programmiert wurde. Rich et al. (2020) haben ebenfalls beschrieben, dass Lernende Variablen zuerst aus Sicht des Computers und erst später aus Sicht einer programmierenden Person betrachten sollten.

Die Scratchprojekte, die im Interview verwendet wurden, haben bei den Lernenden Anklang gefunden. Einige Kinder berichteten nach dem Interview, dass das Spielen ihnen Spass bereitet habe. Insbesondere die drei Programme, die eine alltägliche Situation zeigen sollten, stimmten mit den Beschreibungen der jeweiligen Alltagsvariable in zufriedenstellendem Masse überein. Die erstellten Programme können als Prototypen für weitere Untersuchungen verwendet werden.

7. Diskussion

Die Untersuchungen dieser Arbeit zielten darauf ab, die Vermittlung des Programmierkonzepts ›Variablen‹ für Lernende im Alter von 10–12 Jahren zu analysieren. Um dieses Ziel zu erreichen, wurden Lehrmittel und Forschungsbeiträge untersucht, Expertinnen befragt und Kinder interviewt. Diese Arbeit war entlang der Phasen des iterativen DBR-Prozesses aufgebaut, wodurch sich die Fragestellungen und die Richtung dieser Arbeit durch Ergebnisse während des Forschungsprozesses ergeben haben.

In U1 stand die Analyse von Lehrmitteln und Forschungsbeiträgen im Fokus. Hierbei wurde ermittelt, welche Analogien verwendet werden, um das Programmierkonzept ›Variablen‹ zu erklären. Es wurde unter anderem festgestellt, dass neben der häufig verwendeten – jedoch auch kritisch beurteilten – Box-Analogie auch der Einsatz von Alltagsanalogien möglich sein könnte.

U2 diente dazu, die Meinung von drei Expertinnen auf dem Feld einzubeziehen. Dabei wurden neben möglichen Aktivitäten auch wertvolle didaktische Hinweise zum Informatikunterricht mit Variablen gesammelt. Die Meinungen der Expertinnen zeigen auf, dass keine bisher verwendete Analogie als ideal betrachtet werden kann und dass es vielmehr als sinnvoll erachtet wird, einzelne Eigenschaften des Programmierkonzepts ›Variablen‹ mit einer Analogie zu erklären. Als eine Option wurde die Verwendung von Alltagsvariablen wie Zeit und Temperatur für die Erklärung der Veränderlichkeit der Werte von Variablen vorgeschlagen. Dies wird als eine Möglichkeit betrachtet, um bei der Entwicklung von Unterricht an das von den Expertinnen als äusserst relevant beschriebene Vorwissen der Lernenden anzuknüpfen.

Die in U3 ausgewerteten Beschreibungen von Lernenden im Alter von 10–12 Jahren zu den Alltagsvariablen ›Temperatur‹, ›Zeit‹ und ›Geschwindigkeit‹ wiesen Überschneidungen mit dem Programmierkonzept ›Variablen‹ auf. Genauer beschrieben die Lernenden Temperatur als etwas Veränderliches, was auch auf die Werte von Variablen zutrifft. Auf Grundlage dieses Ergebnisses wird die Hypothese aufgestellt, dass die Alltagsanalogie ›Temperatur‹ im Unterricht mit Lernenden im Alter von 10–12 Jahren zur Erklärung der Veränderlichkeit von Werten einer Variable verwendet werden könnte. Darüber hinaus wurden die Lernenden in diesem Interview darum gebeten, das Programmierkonzept ›Variablen‹ zu beschreiben. Diese meist intuitiven Vorstellungen können als wertvolle Sammlung der für die Unterrichtsentwicklung relevanten Perspektive von Lernenden gesehen werden.

Im Zusammenschluss des theoretischen Hintergrunds mit den Ergebnissen aus U1, U2 und U3 wurden DP erschlossen und werden im Folgenden begründet.

DP1: Für das Programmierkonzept «Variablen» gibt es keine ideale Analogie. Bei der Verwendung von Analogien in einer Unterrichtssequenz über das Programmierkonzept «Variablen» sollten daher Gemeinsamkeiten und Unterschiede zwischen der Analogie und Variablen angesprochen werden.

In der Theorie, den Lehrmitteln und Forschungsbeiträgen sowie laut den Expertinnen sind unterschiedliche Analogien bekannt. Dazu gehören beispielsweise die Box-Analogie, die Label-Analogie, die Whiteboard-Analogie, die Briefumschlags-Analogie oder auch Alltagsanalogien. Die Expertinnen beschrieben, dass keine Analogie eine perfekte Übereinstimmung mit dem Programmierkonzept «Variablen» aufweist. Aus der Theorie über Analogien ist bekannt, dass Eigenheiten vorkommen können (Kircher & Girwidz, 2020), Lernenden jedoch aufgezeigt werden sollen (Aubusson et al., 2006). So wäre es denkbar, Lehrpersonen dazu anzuleiten, Unterricht entlang des «Focus-Action-Reflection»-Leitfadens (Treagust et al., 1998) aufzubauen. Darin nimmt der Vergleich zwischen Analogie und Zielkonzept eine zentrale Rolle ein. In einigen der untersuchten Lehrmittel – genauer: in Begleitbänden für Lehrpersonen – konnten entsprechende Hinweise festgestellt werden.

DP2: In einer Unterrichtssequenz über das Programmierkonzept «Variablen» sollten Analogien verwendet werden, um eine oder wenige Eigenschaften von Variablen zu erklären.

Variablen stellen ein anspruchsvolles Programmierkonzept (Kohn, 2017) dar, das verschiedene Eigenschaften aufweist (Kapitel 2.1.4). Da eine Analogie wie zuvor beschrieben nicht ideal ist, schlagen die Expertinnen aus U2 vor, Analogien zu verwenden, um eine oder wenige Eigenschaften von Variablen zu erklären. Die Box-Analogie wird häufig genutzt, um das Speichern von Werten in einer Variable zu verdeutlichen (Van Der Werf et al., 2023). Die Label-Analogie könnte den Expertinnen zufolge genutzt werden, um das Aktualisieren von Werten zu erläutern. Aus der simplen Analogie «die Variable ist wie eine Box» könnte durch den Fokus auf eine Eigenschaft einer Variable eine angereicherte Analogie «die Variable ist wie eine Box, weil sie jeweils einen Wert speichert» entwickelt werden (Curtis & Reigeluth, 1984).

DP3: Eine Unterrichtssequenz über das Programmierkonzept «Variablen» sollte an das Vorwissen der Lernenden anknüpfen.

Die Perspektive der Lernenden und damit die Vorstellungen, Ideen und Gedanken, die sie in den Informatikunterricht mitbringen, sollten bei der Planung des Unterrichts einbezogen werden (Diethelm et al., 2011; Kattmann et al., 1997). Auch die Expertinnen beschreiben das Vorwissen als relevanten Ausgangspunkt bei der Planung von Informatikunterricht mit Variablen. Geldreich (2023, S. 295) beschreibt als DP in ihrer Dissertation über Programmierunterricht in der Grundschule Folgendes: «Die Unterrichtssequenz führt von Bekannten zum Unbekannten.» Dabei beschreibt die Autorin, dass es relevant ist, an den Alltag und das Vorwissen der Lernenden beispielsweise durch vertraute Beispiele anzuknüpfen. Dies scheint auch für das Programmierkonzept «Variablen» relevant zu sein.

DP4: Eine Unterrichtssequenz über das Programmierkonzept «Variablen» sollte das Bedienen von Programmen oder Spielen mit sichtbaren Variablen beinhalten.

In U3 durften Lernende verschiedene Programme oder Spiele in Scratch anschauen, die eine bis zwei Variablen zeigten. Sie wurden dazu aufgefordert, zu beschreiben, was sie machen oder was sie denken. Die Beschreibungen der Programme und der Variablen während dem Bedienen der Programme wurden zwar nicht im Hinblick auf eine FF ausgewertet. Trotzdem konnten einige Beispiele gezeigt werden, aus denen hervorgeht, dass die gezeigten Programme verstanden wurden. Lernbegleitungsmethoden wie «Use-Modify-Create» fordern auch, dass dem Betrachten und Benutzen von vorgegebenen Programmen im Lernprozess Beachtung geschenkt werden (Lytle et al., 2019). Bevor ein Konzept vereinfacht erklärt oder visualisiert wird, soll es gemäss der Theorie der «Semantic Wave» im technischen Sinne erklärt werden (Waite et al., 2019). Die vorliegenden Untersuchungen zeigen erste Ergebnisse dafür, dass solche Programmierbeispiele mit sichtbaren Variablen einen Einblick für das Programmierkonzept in echten Programmen ermöglichen könnten. Es wird also vermutet, dass diese Beispiele sich für einen Einstieg in den Lernprozess eignen könnten.

DP5: In einer Unterrichtssequenz über das Programmierkonzept «Variablen» kann die Variable «Punktestand» verwendet werden.

Die Analyse der in Lehrmitteln und Forschungsbeiträgen sowie von den Expertinnen erwähnten Aktivitäten hat gezeigt, dass die Variable «Punktestand» häufig thematisiert wird. Auch Untersuchungen von Programmen von Lernenden zeigen, dass diese Punktestandvariable häufiger verwendet wird als andere (Paparo et al., 2021; Vaníček, 2015). In U3 wurde diese Variable in den Spielen teilweise als sichtbare Variable gezeigt. Es erfolgte keine vergleichende Auswertung der Beschreibungen der unterschiedlichen Variablen während des Spielens. Trotzdem wurde beobachtet, dass einige Lernende diese Veränderungen des Wertes der Punktestandvariable erklären konnten. Solchen informatischen Phänomenen, die durch automatisierte Informationsverarbeitung verursacht und von Lernenden beobachtet wurden, sollten in der Entwicklung von Informatikunterricht Beachtung geschenkt werden (Diethelm et al., 2011). Dies ermöglicht die Annahme, dass der Einsatz der Punktestandvariable im Informatikunterricht sinnvoll sein könnte.

DP6: In einer Unterrichtssequenz über das Programmierkonzept «Variablen» kann die Alltagsanalogie «Temperatur» verwendet werden, um die Veränderlichkeit von Werten zu illustrieren.

Die Ergebnisse aus U1 und U2 zeigten auf, dass eine Analogie mit einer Variable aus dem Alltag der Lernenden möglicherweise als Erklärung für Variablen genutzt werden könnte. Die Beschreibungen der Lernenden aus U3 über die Alltagsvariable «Temperatur» zeigten eine eindeutige Überschneidung mit dem Programmierkonzept «Variable». Die Temperatur wurde von einigen Lernenden als veränderbar beschrieben. Auf die Werte von Variablen trifft dies ebenfalls zu (Fischer & Hofer, 2011, S. 956). Diese Ergebnisse legen die Hypothese nahe, dass Eigenschaften des Programmierkonzepts «Variable» Lernenden im Alter von 10–12 Jahren mithilfe der Alltagsanalogie «Temperatur» erklärt werden könnten. Daraus erschliesst sich folgende Formulierung einer angereicherten Analogie: «Eine Variable ist wie die Temperatur, weil sich ihre Werte verändern können».

8. Reflexion und Ausblick

8.1 Implikationen für die Fachdidaktik Informatik

Die Ergebnisse der vorliegenden Untersuchungen leisten einen Beitrag zur Didaktischen Rekonstruktion des Programmierkonzepts *«Variablen»*. Wie in Kapitel 3.4 beschrieben, liefern der theoretische Hintergrund sowie die Ergebnisse aus U1 und U2 Hinweise zur fachlichen Klärung sowie zur gesellschaftlichen Relevanz. Einen ersten Einblick in die Perspektive der Lernenden ermöglichen die Resultate zu U3. Die erfassten Beschreibungen des Programmierkonzepts *«Variablen»* bietet einen Einblick in mögliche Vorstellungen von Lernenden im Alter von 10–12 Jahren. Die Resultate bezüglich der Variable *«Punktestand»* lassen die Vermutung zu, dass dieser vielleicht ein für Lernende beobachtbares informatisches Phänomen sein könnte. Zusammengefasst werden diese Punkte in den beschriebenen DP, welche als grundlegende Leitlinien für zukünftige Unterrichtsentwicklungen über das Programmierkonzept Variable beachtet werden sollten. Somit leisten die Resultate dieser Arbeit einen Beitrag zur didaktischen Strukturierung von Informatikunterricht.

8.2 Diskussion des methodischen Vorgehens

In dieser Masterarbeit wurden mithilfe des DBR-Ansatzes DP für den Informatikunterricht in Bezug auf das Programmierkonzept *«Variablen»* ermittelt. Dieser Forschungsprozess war durch den iterativen Charakter nicht vorab planbar, sondern eher ergebnisgeleitet. Das hat dazu geführt, dass schliesslich drei Untersuchungen durchgeführt wurden. In Bezug auf die gesamte Arbeit lässt sich festhalten, dass der DBR-Prozess noch nicht abgeschlossen werden konnte. Das bedeutet, die verfassten DP wurden weder mehrfach noch in unterschiedlichen Kontexten überprüft. Dies limitiert ihren Wirkungsbereich. Es wird daher vorgeschlagen, diese eher als Hypothesen zu betrachten, die es zukünftig noch zu bestätigen gilt.

Im Nachhinein wird die strikte Trennung von U1 und U2 als eher unpassend eingeschätzt. Die beiden Untersuchungen verliefen zeitgleich und mit ähnlichen FF. Die klare Abgrenzung der beiden Teile hat beim Schreiben der Arbeit dazu geführt, dass einige Aspekte mehrfach beschrieben werden mussten. Eine nähere Betrachtung der beiden Untersuchungen hätte potenziell eine weitere FF zu Gemeinsamkeiten und Unterschieden ermöglicht.

Im Folgenden werden die Limitationen aus U1, U2 und U3 einzeln beschrieben.

8.2.1 Analyse Lehrmittel und Forschungsbeiträge

Der Suche nach Lehrmitteln und Forschungsarbeiten unterlag, wie in Kapitel 4.2 beschrieben, einem strukturierten Vorgehen. Trotzdem konnten nur Beiträge ausgewählt werden, zu denen

ein Zugang möglich war. Damit handelt es sich um eine Gelegenheitsstichprobe (Döring & Bortz, 2016, S. 307). Die Aussagekraft der Ergebnisse von U1 ist dadurch limitiert. Geldreich et al. (2019) beschreiben, dass es im DBR-Prozess keine Richtlinien für diesen Rechercheprozess gibt und deshalb die Ansicht zum Thema von unterschiedlichen Personen mit Expertise auf diesem Gebiet eingeholt werden sollte. Mit U2 wurde dieser Forderung nachgegangen.

Die Analyse der Lehrmittel und Forschungsbeiträge wurde nur von einer Person durchgeführt. Das Kategoriensystem und der Kodierprozess wurden nicht im Sinne einer Triangulation oder mittels Intra- oder Intercoder-Reliabilität getestet. Das schränkt die Reliabilität dieser Untersuchung ein. Um dem entgegenzuwirken, wurde ein Kategoriensystem entwickelt, in dem die Zuordnung zu den einzelnen Ausprägungen beschrieben wurde.

Die Kategorisierung der Rollen von Variablen wurde als anspruchsvoller Prozess empfunden. Zusätzlich wurde nur das einmalige und nicht das vielfache Auftreten der Rolle berücksichtigt, weshalb auch im Ergebnis-Kapitel keine Angaben über die Anzahl gemacht wurden. Diese Resultate müssen somit äusserst kritisch betrachtet werden und im Kontext späterer wissenschaftlicher Arbeiten weiter untersucht werden.

8.2.2 Experten- und Expertinneninterviews

U2 weist mehrere limitierende Faktoren auf. Die bereits kleine Stichprobe wird zusätzlich dadurch geschmälert, dass zwei Expertinnen im selben Forschungsteam gearbeitet haben. Die Antworten dieser beiden Expertinnen haben deshalb gewisse Übereinstimmungen aufgezeigt. Das schränkt die in U2 erfasste Expertise auf eine eher praktische und eine wissenschaftlich geprägte Ansicht anstelle der angenommenen drei Ansichten ein. Es ist deshalb davon auszugehen, dass zu diesem Thema noch andere Meinungen von Expertinnen oder Experten existieren.

Die Interviews mit Expertinnen wurden von nur einer Person geleitet und ausgewertet. Aufgrund begrenzter Ressourcen konnte keine Überprüfung der Intra- oder Interrater-Reliabilität erfolgen, was die Verlässlichkeit der Ergebnisse beeinträchtigt. Ein Kodierleitfaden mit detaillierten Beschreibungen und Ankerbeispielen wurde erstellt, um den Kodierprozess für andere nachvollziehbar zu machen.

Weiter musste festgestellt werden, dass der Begriff «didaktische Hinweise» in FF2 breit gefasst ist. Es konnten in diesem Teil zudem nicht alle Aussagen der Expertinnen präsentiert werden – vielmehr musste eine Auswahl getroffen werden. Daher wurden nur für den Forschungsprozess relevante Ergebnisse berichtet.

Zuletzt gaben alle drei Expertinnen an, dass der letzte Teil des Interviews – das Brainstorming über die Analogien und FV – äusserst anspruchsvoll gewesen sei. Das deutet darauf hin, dass

die Aussagen in diesem Teil eher als Vermutungen angesehen werden können anstatt als fachlich fundierte Aussagen.

8.2.3 Kinderinterviews

Bei dieser Untersuchung handelt es sich um eine explorative Studie mit einer kleinen Stichprobe. Aus diesem Grund können die Ergebnisse nicht auf alle Lernenden im Alter von 10–12 Jahren bezogen interpretiert werden. Zusätzlich wurde das Vorwissen der Lernenden nicht erfasst. Es kann deshalb nicht beurteilt werden, wie diese beschriebenen Vorstellungen der Lernenden zustande gekommen sind sowie ob und wie die gezeigten Scratchprogramme einen Einfluss auf dieses Verständnis gehabt haben.

Die Beurteilung der selbst formulierten Kategorien über die Beschreibungen des Programmierkonzepts «Variablen» muss kritisch betrachtet werden. Bei der Formulierung der Kategorien wurde darauf geachtet, Wortlaute der Lernenden zu übernehmen, jedoch handelt es sich unabhängig davon um verallgemeinernde Aussagen. Diese Formulierungen sind potenziell durch subjektive Interpretation verzerrt. Um genauere Aussagen über die Art der Vorstellungen der Lernenden in diesem Alter zu erhalten, müssten deshalb Beschreibungen der Lernenden selbst genauer in den Blick genommen werden.

Die Auswirkungen der unterschiedlichen Hinweisstufen auf die Antworten der Lernenden wurden nicht ausgewertet. Das bedeutet, es geht aus den Ergebnissen der Arbeit nicht hervor, welchen Einfluss diese Hinweise auf die Beschreibung der Lernenden gehabt haben.

Die Interviews mit Kindern in U3 wurden ausschliesslich von einer Person durchgeführt und ausgewertet. Aus forschungsökonomischen Gründen musste auf eine Intra- oder Interraterreliabilität verzichtet werden, was die Verlässlichkeit der Resultate einschränkt. Der Kodierleitfaden mit Beschreibung und Ankerbeispiel der Ausprägung soll helfen, den Kodierprozess dieser Untersuchung für andere nachvollziehbar zu machen.

Die Interpretation von offenen Fragen ist anspruchsvoll und nicht frei von subjektiven Einflüssen (Döring & Bortz, 2016). Bei Interviews mit Kindern kommt dazu, dass diese von erwachsenen Personen häufig unterschätzt werden, weil sie andere Formulierungen verwenden (Vogl, 2021). Aus diesem Grund wurden, wann immer möglich, Antworten von Lernenden im Ergebnisstil zitiert.

8.3 Zukünftige Forschung und Empfehlungen

Diese Arbeit bietet einige Anknüpfungspunkte für zukünftige Forschungen, die im Folgenden kurz vorgestellt werden.

- DBR: Die Ergebnisse dieser Arbeit können genutzt werden, um eine prototypische Intervention zum Programmierkonzept «Variablen» zu erstellen, die anschliessend im Rahmen mehrerer Untersuchungen getestet werden kann. Neben den DP können für die Entwicklung die gesammelten Aktivitäten als Ideenpool herangezogen werden. Zusätzlich könnte auch die Eignung der entwickelten Scratchprogramme analysiert werden.
- Alltagsanalogie Variable: Um mehr Erkenntnisse über die Eignung der Alltagsanalogie «Temperatur» zu erhalten, muss der Einsatz dieser weiter erforscht werden. Dazu kann beispielsweise mittels Post- und Pretest-Verfahren der Lernzuwachs in Bezug auf das Programmierkonzept «Variablen» mittels geeigneter Testinstrumente festgestellt werden.
- Didaktische Rekonstruktion: Die Didaktische Rekonstruktion zeigt auf, dass es bedeutend ist, die Perspektive von Lehrpersonen zu erfassen. Dazu wäre eine Befragung mit Lehrpersonen über ihre Vorstellung vom Programmierkonzept «Variablen» sinnvoll. Zudem gilt es, die Eignung des Punktestands als informatisches Phänomen zu klären.
- Vorstellungen von Lernenden: Die in U3 gesammelten Beschreibungen von Lernenden zum Programmierkonzept «Variablen» sind interessant, aber noch nicht vollständig oder gar repräsentativ erfasst. Um genauere Ergebnisse zu intuitiven und elaborierten Vorstellungen von Lernenden zu erhalten, müssten nicht nur die Kategorien, sondern auch die Beschreibungen der Lernenden genau untersucht werden.

8.4 Fazit

In der vorliegenden Masterarbeit wurde der Informatikunterricht für 10–12-jährige Lernende in Bezug auf das Programmierkonzept «Variablen» in den Blick genommen. Erste Ergebnisse deuten darauf hin, dass Analogien zur Erklärung von Variablen genutzt werden können. Es sollte jedoch genau analysiert werden, welche Übereinstimmungen und Abweichungen zwischen Analogie und Variable bestehen. Für den Informatikunterricht eignen sich Analogien, um eine oder wenige Eigenschaften von Variablen zu erklären. Die Analogie sollte dabei aus der Lebenswelt der Kinder stammen und auf ihrem Vorwissen aufbauen. Es wurden Hinweise dafür entdeckt, dass die Alltagsvariable «Temperatur» als Analogie zur Erklärung der Veränderlichkeit der Werte von Variablen genutzt werden könnte. Zusätzlich konnten erste, meist intuitive Vorstellungen über Variablen von Lernenden im Alter von 10–12 Jahren gesammelt werden, indem ihre Beschreibungen nach dem Spielen einfacher Scratchprogramme mit sichtbaren Variablen erhoben wurden.

9. Literaturverzeichnis

- Affolter, W., Amstad, H., Doebeli, M., & Wieland, G. (2018). *Schweizer Zahlenbuch 6—Neue Ausgabe*. Klett und Balmer Verlag.
- Analogy-Based Notional Machines*. (o. J.). Notional Machines. Abgerufen 25. Juli 2024, von <https://notionalmachines.github.io/analogy.html>
- Anderson, T., & Shattuck, J. (2012). Design-Based Research: A Decade of Progress in Education Research? *Educational Researcher*, 41(1), 16–25. <https://doi.org/10.3102/0013189X11428813>
- Aubusson, P., Harrison, A. G., & Ritchie, S. (2006). Metaphor and Analogy. In P. Aubusson, A. G. Harrison, & S. Ritchie (Hrsg.), *Metaphor and analogy in science education* (S. 1–9). Springer.
- Babari, P., Hielscher, M., Edelsbrunner, P. A., Conti, M., Honegger, B. D., & Marinus, E. (2023). A literature review of children's and youth's conceptions of the internet. *International Journal of Child-Computer Interaction*, 37, 100595. <https://doi.org/10.1016/j.ijcci.2023.100595>
- Bagge, P., & Grover, S. (2020). Variables. In S. Grover (Hrsg.), *Computer Science in K-12: An A to Z handbook on teaching programming* (S. 227–239). Edfinity.
- Barab, S., & Squire, K. (2004). Design-Based Research: Putting a Stake in the Ground. *Journal of the Learning Sciences*, 13(1), 1–14. https://doi.org/10.1207/s15327809jls1301_1
- Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4), 499–528. <https://doi.org/10.1007/s40692-019-00147-3>
- Bevans, R. (2022, September 19). *Types of Variables in Research & Statistics | Examples*. Scribbr. <https://www.scribbr.com/methodology/types-of-variables/>
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72. <https://doi.org/10.1145/3137065.3137069>
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*.
- Brookshear, J. G. (2012). *Computer science: An overview* (11. Auflage). Addison-Wesley.
- Buffum, P. S., Ying, K. M., Zheng, X., Boyer, K. E., Wiebe, E. N., Mott, B. W., Blackburn, D. C., & Lester, J. C. (2018). Introducing the Computer Science Concept of Variables in Middle School Science Classrooms. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 906–911. <https://doi.org/10.1145/3159450.3159545>
- Ching, Y.-H., Hsu, Y.-C., & Baldwin, S. (2018). Developing Computational Thinking with Educational Technologies for Young Learners. *TechTrends*, 62(6), 563–573. <https://doi.org/10.1007/s11528-018-0292-7>
- Choi, H., Kim, E., & Paik, S. (2001). *Investigating Elementary Students Understanding Levels and Alternative Conceptions Of Heat And Temperature*.
- Claus, V., & Schwill, A. (2006). Variable. In *Duden Informatik A - Z: Fachlexikon für Studium, Ausbildung und Beruf* (4. Aufl, S. 717–719). Dudenverl.

- Corney, M., Lister, R., & Teague, D. (2011). Early Relational Reasoning and the Novice Programmer: Swapping as the «Hello World» of Relational Reasoning. *Conferences in Research and Practice in Information Technology Series*, 114, 95–104.
- Curtis, R. V., & Reigeluth, C. M. (1984). The Use Of Analogies In Written Text. *Instructional Science*, 13(2), 99–117.
- Curzon, P., McOwan, P. W., Donohue, J., Wright, S., & Mars, D. W. (2018). Teaching of concepts. In S. Sentence, E. Barensen, & C. Schulte (Hrsg.), *Computer Science Education: Perspectives on Teaching and Learning in School* (S. 91–108). Bloomsbury Publishing.
- D-EDK (Hrsg.). (2016a). *Lehrplan 21—Medien und Informatik*. https://v-fe.lehrplan.ch/container/V_FE_DE_Modul_MI.pdf
- D-EDK (Hrsg.). (2016b). *Lehrplan 21—Natur, Mensch, Gesellschaft*. https://v-fe.lehrplan.ch/container/V_FE_DE_Fachbereich_NMG.pdf
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832. <https://doi.org/10.1016/j.compedu.2020.103832>
- Denz, H. (1989). *Einführung in die empirische Sozialforschung*. Springer Vienna. <https://doi.org/10.1007/978-3-7091-9067-8>
- Diethelm, I., & Dörge, C. (2011). Zur Diskussion von Kontexten und Phänomenen in der Informatikdidaktik. In M. Thomas (Hrsg.), *Informatik in Bildung und Beruf: 14. GI-Fachtagung «Informatik und Schule—INFOS 2011», 12. - 15. September 2011 an der Westfälischen Wilhelms-Universität Münster* (S. 76). Ges. für Informatik.
- Diethelm, I., Dörge, C., Mesaros, A.-M., & Dunnebier, M. (2011). Die Didaktische Rekonstruktion für den Informatikunterricht. In *Informatik in Bildung und Beruf—14. GI-Fachtagung «Informatik und Schule—INFOS 2011»* (S. 77–86). <https://dl.gi.de/handle/20.500.12116/18479>
- Diethelm, I., Hubwieser, P., & Klaus, R. (2012). Students, teachers and phenomena: Educational reconstruction for computer science education. *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*, 164–173. <https://doi.org/10.1145/2401796.2401823>
- Döring, N., & Bortz, J. (2016). *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-41089-5>
- Dresing, T., & Pehl, T. (2018). *Praxisbuch Interview, Transkription & Analyse: Anleitungen und Regelsysteme für qualitativ Forschende* (8. Auflage). Eigenverlag.
- Du Boulay, B. (1986). Some Difficulties of Learning to Program. *Journal of Educational Computing Research*, 2(1), 57–73. <https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9>
- Duit, R. (1991). On the role of analogies and metaphors in learning science. *Science Education*, 75(6), 649–672. <https://doi.org/10.1002/sce.3730750606>
- Easterday, M. W., Lewis, D. R., & Gerber, E. M. (2014). *Design-Based Research Process: Problems, Phases, and Applications*.
- Edelsbrunner, P. A., Schalk, L., Schumacher, R., & Stern, E. (2018). Variable control and conceptual change: A large-scale quantitative study in elementary school. *Learning and Individual Differences*, 66, 38–53. <https://doi.org/10.1016/j.lindif.2018.02.003>

- Ely, R., & Adams, A. E. (2012). Unknown, placeholder, or variable: What is x? *Mathematics Education Research Journal*, 24(1), 19–38. <https://doi.org/10.1007/s13394-011-0029-9>
- Euler, D. (2014). Design-Research—A paradigm under development. In D. Euler & P. F. E. Sloane (Hrsg.), *Design-based research* (S. 15–44). Franz Steiner Verlag.
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jr. On the iPad. *Journal of Computer Assisted Learning*, 32(6), 576–593. <https://doi.org/10.1111/jcal.12155>
- Fikisz, W., Bucher, K., Ciprina, H. J., Meyer, M., Rottinger, U., & Schmid, J. (2023). *vernetzt—Digitale Grundbildung* (1. Auflage). Österreichischer Bundesverlag Schulbuch. <https://www.oebv.at/flippingbook/9783209128294/II/>
- Fincher, S., Jeuring, J., Miller, C. S., Donaldson, P., Du Boulay, B., Hauswirth, M., Hellas, A., Hermans, F., Lewis, C., Mühling, A., Pearce, J. L., & Petersen, A. (2020). Notional Machines in Computing Education: The Education of Attention. *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*, 21–50. <https://doi.org/10.1145/3437800.3439202>
- Fischer, P., & Hofer, P. (2011). *Lexikon der Informatik*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-15126-2>
- Fischler, H., & Schecker, H. (2018). Schülervorstellungen zu Teilchen und Wärme. In H. Schecker, T. Wilhelm, M. Hopf, & R. Duit (Hrsg.), *Schülervorstellungen und Physikunterricht* (S. 139–161). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-57270-2_7
- Friedman, W. J. (1982). Conventional time concepts and children's structuring of time. *The developmental psychology of time*, 171–208.
- Geldreich, K. (2023). *Programmieren in der Grundschule. Eine Design-Based-Research-Studie* [Dissertation, Technischen Universität München]. <https://media-tum.ub.tum.de/doc/1703768/1703768.pdf>
- 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, 53–75. <https://doi.org/10.21240/mpaed/33/2019.02.15.X>
- Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy*. *Cognitive Science*, 7(2), 155–170. https://doi.org/10.1207/s15516709cog0702_3
- Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Hrsg.), *Similarity and Analogical Reasoning* (1. Aufl., S. 199–241). Cambridge University Press. <https://doi.org/10.1017/CBO9780511529863.011>
- Geschwindigkeit. (2024). In *Wikipedia*. <https://de.wikipedia.org/w/index.php?title=Geschwindigkeit&oldid=246472832>
- Gisler, L., & Regli, N. (2023). *Förderung der Programmierkompetenzen mit ScratchJr in der 1./2. Klasse* [Unveröffentlichter Leistungsnachweis MSWT]. PHSZ.
- Glynn, S. M., Britton, B. K., Semrud-Clikeman, M., & Muth, K. D. (1989). Analogical Reasoning and Problem Solving in Science Textbooks. In J. A. Glover, R. R. Ronning, & C. R. Reynolds

- (Hrsg.), *Handbook of Creativity* (S. 383–398). Springer US. https://doi.org/10.1007/978-1-4757-5356-1_21
- Glynn, S. M., & Duit, R. (1995). Learning Science Meaningfully: Construction Conceptual Models. In S. M. Glynn & R. Duit (Hrsg.), *Learning science in the schools: Research reforming practice* (S. 3–34). L. Erlbaum Associates.
- Grover, S., & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 267–272. <https://doi.org/10.1145/3017680.3017723>
- Grover, S., Jackiw, N., & Lundh, P. (2019). Concepts before coding: Non-programming interactives to advance learning of introductory programming concepts in middle school. *Computer Science Education*, 29(2–3), 106–135. <https://doi.org/10.1080/08993408.2019.1568955>
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Grover, S., & Pea, R. D. (2018). Computational Thinking: A Competency Whose Time Has Come. In S. Sentance, E. Barendsen, & C. Schulte (Hrsg.), *Computer Science Education: Perspectives on Teaching and Learning in School* (S. 19–38). Bloomsbury Academic. <https://doi.org/10.5040/9781350057142>
- Guzdial, M. (2016). *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-02216-6>
- Haider, M. (2010). *Der Stellenwert von Analogien für den Erwerb naturwissenschaftlicher Erkenntnisse: Eine Untersuchung im Sachunterricht der Grundschule am Beispiel «Elektrischer Stromkreis»*. Verlag Julius Klinkhardt.
- Haider, M., & Fölling-Albers, M. (2020). Auswirkungen von Analogiemodellen auf den Aufbau konzeptuellen Wissens im Sachunterricht der Grundschule – Beispiel Stromkreis. *Unterrichtswissenschaft*, 48(3), 469–491. <https://doi.org/10.1007/s42010-020-00077-5>
- Harrison, A. G., & Treagust, D. F. (1993). Teaching with analogies: A case study in grade-10 optics. *Journal of Research in Science Teaching*, 30(10), 1291–1307. <https://doi.org/10.1002/tea.3660301010>
- Harrison, A. G., & Treagust, D. F. (2000). Learning about atoms, molecules, and chemical bonds: A case study of multiple-model use in grade 11 chemistry. *Science Education*, 84(3), 352–381. [https://doi.org/10.1002/\(SICI\)1098-237X\(200005\)84:3<352::AID-SCE3>3.0.CO;2-J](https://doi.org/10.1002/(SICI)1098-237X(200005)84:3<352::AID-SCE3>3.0.CO;2-J)
- Hartmann, M., Edelsbrunner, P. A., Hielscher, M., Paparo, G., Döbeli Honegger, B., & Marinus, E. (2022). Programming concepts and misconceptions in grade 5 and 6 children: Developing and testing a new assessment tool. *Atti del 5° Convegno sulle didattiche disciplinari*, 328–333. <https://doi.org/10.33683/dida.22.05.59>
- Hauswirth, M. (2024, Februar 19). *Variable as Pigeonhole*. Notional Machines. <https://notionalmachines.github.io/nms/VariableAsPigeonhole.html>
- Havenga, M., Breed, B., Mentz, E., Govender, I., Dignum, F., & Dignum, V. (2013). Metacognitive and Problem-Solving Skills to Promote Self-Directed Learning in Computer Programming: Teachers' Experiences. *SA-eDUC Journal*, 10(2), 1–14.

- Helfferrich, C. (2014). Leitfaden- und Experteninterviews. In N. Baur & J. Blasius (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung* (S. 559–574). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-531-18939-0_39
- Hermans, F. (o. J.). *Variable as Box* [A Curated Collection]. Notional Machines. Abgerufen 14. Juli 2024, von <https://notionalmachines.github.io/nms/VariableAsBox.html>
- Hermans, F. (2019). Forms Of Notional Machines. *Felienne Hermans*. <https://www.felienne.com/archives/6392>
- Hermans, F. (2024, Februar 19). *Variable as Clothespin*. Notional Machines. <https://notionalmachines.github.io/nms/VariableAsClothespin.html>
- Hermans, F., Swidan, A., Aivaloglou, E., & Smit, M. (2018). Thinking out of the box: Comparing metaphors for variables in programming education. *Proceedings of the 13th Workshop in Primary and Secondary Computing Education*, 1–8. <https://doi.org/10.1145/3265757.3265765>
- Hielscher, M., & Döbeli Honegger, B. (2015). *Lernumgebungen für den Einstieg ins Programmieren: Versuch einer Klassifikation*. https://programmingwiki.de/images/a/a9/Programmierungsumgebungen_2015.pdf
- Hoffmann, N. (2018). *Dokumentenanalyse in der Bildungs- und Sozialforschung: Überblick und Einführung* (1. Auflage). Beltz Juventa.
- Hoodless, P. A. (2002). An Investigation into Children's Developing Awareness of Time and Chronology in Story. *Journal of Curriculum Studies*, 34(2), 173–200. <https://doi.org/10.1080/00220270110080962>
- Hromkovič, J., & Kohn, T. (2018). *Einfach Informatik 7-9. Programmieren: Begleitband* (1. Auflage). Klett und Balmer Verlag.
- Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M. N., Knobelsdorf, M., Magenheimer, J., Mittermeir, R., & Schubert, S. (2011). Computer science/informatics in secondary education. *Proceedings of the 16th Annual Conference Reports on Innovation and Technology in Computer Science Education - Working Group Reports*, 19–38. <https://doi.org/10.1145/2078856.2078859>
- Kalogiannakis, M., & Papadakis, S. (2018). A proposal for teaching ScratchJR programming environment in preservice kindergarten teachers. In O. Finlayson, E. McLoughlin, S. Erduran, & P. Childs (Hrsg.), *Electronic Proceedings of the ESERA 2017 Conference. Research, Practice and Collaboration in Science Education* (S. 2095–2105). Dublin City University.
- Kattmann, U., Duit, R., Gropengiesser, H., & Komorek, M. (1997). *Das Modell der Didaktischen Rekonstruktion – Ein Rahmen für naturwissenschaftsdidaktische Forschung und Entwicklung*. 3, 3–18.
- Kircher, E. (1995). Analogien im Sachunterricht der Primarstufe. *Sachunterricht und Mathematik in der Primarstufe*, 23(5), 192–198.
- Kircher, E., & Girwidz, R. (2020). Elementarisierung und didaktische Rekonstruktion. In E. Kircher, R. Girwidz, & H. E. Fischer (Hrsg.), *Physikdidaktik | Grundlagen* (S. 155–197). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-59490-2_5
- Koh, H., & Paik, S. (2002). Analysis of conceptions of heat and temperature of the pre-service elementary school teachers. *Elementary Science Education*, 21, 81–100.

- Kohn, T. (2017). Variable Evaluation: An Exploration of Novice Programmers' Understanding and Common Misconceptions. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 345–350. <https://doi.org/10.1145/3017680.3017724>
- Kohn, T., & Komm, D. (2019). Denn sie wissen nicht, was sie programmieren. *Informatik Spektrum*, 42(2), 112–117. <https://doi.org/10.1007/s00287-019-01157-2>
- Kommission für Lehrplan- und Lehrmittelfragen. (2023). *Lehrmittelverzeichnis für das Schuljahr 2024-2025*. Bildungs- und Kulturdirektion Bern. <https://www.faecher-net.bkd.be.ch/de/start/themen/lehrmittelverzeichnis.html>
- Kuckartz, U., & Rädiker, S. (2022). *Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung: Grundagentexte Methoden* (5. Auflage). Beltz Juventa.
- Kuittinen, M., & Sajaniemi, J. (2004). Teaching roles of variables in elementary programming courses. *ITiCSE'04*.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37. <https://doi.org/10.1145/1929887.1929902>
- Leppink, J., Paas, F., Van Gog, T., Van Der Vleuten, C. P. M., & Van Merriënboer, J. J. G. (2014). Effects of pairs of problems and examples on task performance and different types of cognitive load. *Learning and Instruction*, 30, 32–42. <https://doi.org/10.1016/j.learninstruc.2013.12.001>
- Linn, M. C., & Songer, N. B. (1991). Cognitive and Conceptual Change in Adolescence. *American Journal of Education*, 99(4), 379–417. <https://doi.org/10.1086/443991>
- Lister, R., Fidge, C., & Teague, D. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, 161–165. <https://doi.org/10.1145/1562877.1562930>
- Lopez, M., Whalley, J., Robbins, P., & Lister, R. (2008). Relationships between reading, tracing and writing skills in introductory programming. *Proceedings of the Fourth International Workshop on Computing Education Research*, 101–112. <https://doi.org/10.1145/1404520.1404531>
- Luo, F., Yan, W., Liu, R., & Israel, M. (2022). Elementary Students' Understanding of Variables in Computational Thinking-Integrated Instruction: A Mixed Methods Study. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, 523–529. <https://doi.org/10.1145/3478431.3499323>
- Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E., & Barnes, T. (2019). Use, Modify, Create: Comparing Computational Thinking Lesson Progressions for STEM Classes. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 395–401. <https://doi.org/10.1145/3304221.3319786>
- Macrides, E., Miliou, O., & Angeli, C. (2022). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction*, 32. <https://doi.org/10.1016/j.ijcci.2021.100396>

- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, 367–371. <https://doi.org/10.1145/1352135.1352260>
- Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, 7(1), 19. <https://doi.org/10.1186/s40594-020-00222-7>
- Maton, K. (2013). Making semantic waves: A key to cumulative knowledge-building. *Linguistics and Education*, 24(1), 8–22. <https://doi.org/10.1016/j.linged.2012.11.005>
- McKenzie, C. (2021). *Pascal case*. TheServerSide - Your Enterprise Java Community. <https://www.theserverside.com/definition/Pascal-case#:~:text=Pascal%20case%20%2D%2D%20or%20PascalCase,names%20to%20include%20blank%20spaces>.
- Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, 62(2), 77–90. <https://doi.org/10.1109/TE.2018.2864133>
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (Moti). (2013). Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 239–264. <https://doi.org/10.1080/08993408.2013.832022>
- Meyer-Baron, L. (2024). *Lernbegleitung im Programmierunterricht. Replikationsstudie zur Wirkung von TIPP&SEE, einer metakognitiven Begleitmethode fürs Programmierenlernen* [Masterarbeit, PHSZ]. <https://zenodo.org/doi/10.5281/zenodo.10785050>
- Miller, C. S. (2024, Februar 19). *Variable Trace Table*. Notional Machines. <https://notionalmachines.github.io/nms/VariableTraceTable.html>
- Munasinghe, B., Bell, T., & Robins, A. (2023). Unplugged activities as a catalyst when teaching introductory programming. *Journal of Pedagogical Research*, 7(2), 55–71. <https://doi.org/10.3390/JPR.202318546>
- Pahl, A., Fuchs, H. U., & Corni, F. (2022). Young Children’s Ideas about Heat Transfer Phenomena. *Education Sciences*, 12(4), Article 4. <https://doi.org/10.3390/educsci12040263>
- Paik, S., Cho, B., & Go, Y. (2007). Korean 4- to 11-year-old student conceptions of heat and temperature. *Journal of Research in Science Teaching*, 44(2), 284–302. <https://doi.org/10.1002/tea.20174>
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187. <https://doi.org/10.1504/IJMLO.2016.077867>
- Paparo, G., Hartmann, M., & Grillenberger, M. (2021). A Scratch Challenge: Middle School Students Working with Variables, Lists and Procedures. *Proceedings of the 21st Koli Calling International Conference on Computing Education Research*, 1–10. <https://doi.org/10.1145/3488042.3488065>
- Pea, R. D. (1986). Language-Independent Conceptual “Bugs” in Novice Programming. *Journal of Educational Computing Research*, 2(1), 25–36. <https://doi.org/10.2190/689T-1R2A-X4W4-29J2>

- Pearce, J. (2024, Februar 19). *Variable as a Parking Space*. Notional Machines. <https://notionalmachines.github.io/nms/VariableAsParkingSpace.html>
- Plaza, P., Sancristobal, E., Carro, G., Castro, M., Blázquez, M., Muñoz, J., & Álvarez, M. (2018). Scratch as Educational Tool to Introduce Robotics. In M. E. Auer, D. Guralnick, & I. Simonics (Hrsg.), *Teaching and Learning in a Digital World* (Bd. 715, S. 3–14). Springer International Publishing. https://doi.org/10.1007/978-3-319-73210-7_1
- Plomp, T. (2013). Educational Design Research: An Introduction. In T. Plomp & N. Nieveen (Hrsg.), *Educational Design Research. Part A: An introduction* (S. 10–51). SLO. Netherlands institute for curriculum development.
- Qian, Y., & Lehman, J. (2018). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Transactions on Computing Education*, 18(1), 1–24. <https://doi.org/10.1145/3077618>
- Reinders, H. (2016). *Qualitative Interviews mit Jugendlichen führen: Ein Leitfaden*. De Gruyter. <https://doi.org/10.1515/9783110469561>
- Rich, K. M., Franklin, D., Strickland, C., Isaacs, A., & Eatinger, D. (2020). A Learning Trajectory for Variables Based in Computational Thinking Literature: Using Levels of Thinking to Develop Instruction. *Computer Science Education*, 32(2), 213–234. <https://doi.org/10.1080/08993408.2020.1866938>
- Rich, K. M., Franklin, D., Strickland, C., Isaacs, A., & Eatinger, D. (2022). A Learning Trajectory for Variables Based in Computational Thinking Literature: Using Levels of Thinking to Develop Instruction. *Computer Science Education*, 32(2), 213–234. <https://doi.org/10.1080/08993408.2020.1866938>
- Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2018). K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. *ACM Inroads*, 9(1), 46–55. <https://doi.org/10.1145/3183508>
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137–172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Rumelhart, D. E., & Norman, D. A. (1981). Analogical Processes in Learning. In J. R. Anderson (Hrsg.), *Cognitive skills and their acquisition* (S. 335–359). Erlbaum.
- Sajaniemi, J. (2002). An empirical analysis of roles of variables in novice-level procedural programs. *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, 37–39. <https://doi.org/10.1109/HCC.2002.1046340>
- Sajaniemi, J., Ben-Ari, M., Byckling, P., Gerdt, P., & Kulikova, Y. (2006). Roles of variables in three programming paradigms. *Computer Science Education*, 16(4), 261–279. <https://doi.org/10.1080/08993400600874584>
- Sajaniemi, J., & Kuittinen, M. (2005). An Experiment on Using Roles of Variables in Teaching Introductory Programming. *Computer Science Education*, 15(1), 59–82. <https://doi.org/10.1080/08993400500056563>
- Sajaniemi, J., & Kuittinen, M. (2006). *An Introduction to the Roles of Variables*. The Roles of Variables Home Page. https://saja.kapsi.fi/var_rols/index.html

- Salac, J., Thomas, C., Butler, C., & Franklin, D. (2021). Investigating the Role of Cognitive Abilities in Computational Thinking for Young Learners. *Proceedings of the 17th ACM Conference on International Computing Education Research*, 2–17. <https://doi.org/10.1145/3446871.3469746>
- Saxena, P., Singh, S. K., & Gupta, G. (2023). Achieving Effective Learning Outcomes through the Use of Analogies in Teaching Computer Science. *Mathematics*, 11(15), 3340. <https://doi.org/10.3390/math11153340>
- Schecker, H., & Wilhelm, T. (2018). Schülervorstellungen in der Mechanik. In H. Schecker, T. Wilhelm, M. Hopf, & R. Duit (Hrsg.), *Schülervorstellungen und Physikunterricht* (S. 63–88). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-57270-2_4
- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 59–66. <https://doi.org/10.1145/2493394.2493403>
- Solé, G. (2019). Children’s understanding of time: A study in a primary history classroom. *History Education Research Journal*, 16(1). <https://doi.org/10.18546/HERJ.16.1.13>
- Sorva, J. (2013). Notional machines and introductory programming education. *ACM Transactions on Computing Education*, 13(2), 1–31. <https://doi.org/10.1145/2483710.2483713>
- Sorva, J. (2018). Misconception and the Beginner Programmer. In S. Sentance, E. Barendsen, & C. Schulte (Hrsg.), *Computer Science Education: Perspectives on Teaching and Learning in School* (S. 171–188). Bloomsbury Academic. <https://doi.org/10.5040/9781350057142>
- Stechert, P. (Regisseur). (2020a). *Die Rollen von Variablen beim Programmier-Unterricht / Informatikdidaktik kurz gefasst Teil 18* [Video recording]. Youtube. https://www.youtube.com/watch?v=wA3tfMNUW_Q
- Stechert, P. (Regisseur). (2020b). *Fehlvorstellungen und Modelle bei Variablen / Informatikdidaktik kurz gefasst Teil 15* [Video recording]. Youtube. <https://www.youtube.com/watch?v=uqTvW5ZhxPc>
- Stechert, P. (Regisseur). (2020c). *Notional Machines als Erklärmodell beim Programmieren / Informatikdidaktik kurz gefasst Teil 19* [Video recording]. Youtube. https://www.google.com/search?q=unterschied+zwischen+analogy+and+notional+machine&oq=unterschied+zwischen+analogy+and+notional+machine+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIG-CAEQLhhA0gEIODk4OWowajSoAgCwAgA&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:32cfd365,vid:w8VMEqqQuig,st:0
- Steffensky, M. (2018). *Frühe naturwissenschaftliche Bildung*. https://www.kita-fachtexte.de/fileadmin/Redaktion/Publikationen/KiTaFT_Steffensky_2018-Fruehe_naturwissenschaftliche_Bildung.pdf
- Stokes, D. E. (1997). *Pasteur’s quadrant: Basic science and technological innovation*. Brookings Institution Press.
- Stow, W. (1999). Issues in the teaching of chronology. In J. Arthur & R. Phillips (Hrsg.), *Issues in history teaching* (S. 83–89). Routledge.

- Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity*, 42, 100926. <https://doi.org/10.1016/j.tsc.2021.100926>
- Treagust, D. F., Harrison, A. G., & Venville, G. J. (1998). Teaching Science Effectively With Analogies: An Approach for Preservice and Inservice Teacher Education. *Journal of Science Teacher Education*, 9(2), 85–101. <https://doi.org/10.1023/A:1009423030880>
- Van Der Werf, V., Zhang, M. Y., Aivaloglou, E., Hermans, F., & Specht, M. (2023). Variables in Practice. An Observation of Teaching Variables in Introductory Programming MOOCs. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 208–214. <https://doi.org/10.1145/3587102.3588857>
- Vaniček, J. (2015). Programming in Scratch Using Inquiry-Based Approach. In A. Brodnik & J. Vahrenhold (Hrsg.), *Informatics in Schools. Curricula, Competences, and Competitions* (Bd. 9378, S. 82–93). Springer International Publishing. https://doi.org/10.1007/978-3-319-25396-1_8
- Vogl, S. (2021). Mit Kindern Interviews führen: Ein praxisorientierter Überblick. In I. Hedderich, J. Reppin, & C. Butschi (Hrsg.), *Perspektiven auf Vielfalt in der frühen Kindheit. Mit Kindern Diversität erforschen* (S. 142–157). Verlag Julius Klinkhardt. <https://doi.org/10.35468/5895-08>
- Waguespack, L. J. (1989). Visual metaphors for teaching programming concepts. *ACM SIGCSE Bulletin*, 21(1), 141–145. <https://doi.org/10.1145/65294.71203>
- Waite, J., Maton, K., Curzon, P., & Tuttiett, L. (2019). Unplugged Computing and Semantic Waves: Analysing Crazy Characters. *Proceedings of the 1st UK & Ireland Computing Education Research Conference on - UKICER*, 1–7. <https://doi.org/10.1145/3351287.3351291>
- Wang, H.-Y., Huang, I., & Hwang, G.-J. (2016). Comparison of the effects of project-based computer programming activities between mathematics-gifted students and average students. *Journal of Computers in Education*, 3(1), 33–45. <https://doi.org/10.1007/s40692-015-0047-9>
- Wehr, L. (2014). Leitfadengestützte Interviews mit Kindern. In C. Bischoff, K. Oehme-Jüngling, & W. Leimgruber (Hrsg.), *Methoden der Kulturanthropologie* (S. 143–158). Haupt.
- Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children*, 199–208. <https://doi.org/10.1145/2771839.2771860>
- Wieczorek, B., Ribe, L., Class, C. B., & Brinkmeier, M. (2017). Analogien für Programmierkonzepte: Ein Weg zum Computational Thinking. *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*, 349–358.
- Wiegand, R. P., Bucci, A., Kumar, A. N., Albert, J. L., & Gaspar, A. (2016). A Data-Driven Analysis of Informatively Hard Concepts in Introductory Programming. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 370–375. <https://doi.org/10.1145/2839509.2844629>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.
- Yusoff, K. M., Sahari, N., Siti, T., & Mohd, N. (2020). Analysis on the Requirements of Computational Thinking Skills to Overcome the Difficulties in Learning Programming. *International Journal of Advanced Computer Science and Applications*, 11(3). <https://doi.org/10.14569/IJACSA.2020.0110329>

- Žanko, Ž., Mladenović, M., & Boljat, I. (2019). Misconceptions about variables at the K-12 level. *Education and Information Technologies*, 24(2), 1251–1268. <https://doi.org/10.1007/s10639-018-9824-1>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>
- Zhang, M., & Hudson, J. A. (2018). Children's understanding of yesterday and tomorrow. *Journal of Experimental Child Psychology*, 170, 107–133. <https://doi.org/10.1016/j.jecp.2018.01.010>

10. Anhangsverzeichnis

10. Anhangsverzeichnis	99
10.1 Anhang U1	101
10.1.1 Stichprobe U1	101
10.1.2 Kriterienraster U1	104
10.1.3 Auswertung U1.....	107
10.2 Anhang U2.....	108
10.2.1 Anfragetext U2	108
10.2.2 Einverständniserklärung deutsch U2	109
10.2.3 Einverständniserklärung englisch U2	110
10.2.4 Interviewleitfaden deutsch U2	111
10.2.5 Interviewleitfaden englisch U2.....	113
10.2.6 Präsentation Interview U2	115
10.2.7 Kodierleitfaden U2.....	118
10.2.8 Interviewtranskript 1	123
10.2.9 Interviewtranskript 2	131
10.2.10 Interviewtranspirt 3	141
10.3 Anhang U3.....	150
10.3.1 Einverständniserklärung U3	150
10.3.2 Interviewleitfaden U3.....	152
10.3.3 Beschreibung der Scratchprogramme	157
10.3.4 Stichprobe U3	159
10.3.5 Kodierleitfaden U3.....	160
10.3.6 Interviewtranskript 1	165
10.3.7 Interviewtranskript 2	174
10.3.8 Interviewtranskript 3	184

10.3.9 Interviewtranskript 4	192
10.3.10 Interviewtranskript 5.....	200
10.3.11 Interviewtranskript 6.....	208
10.3.12 Interviewtranskript 7.....	217
10.3.13 Interviewtranskript 8.....	226
10.3.14 Interviewtranskript 9.....	235
10.3.15 Interviewtranskript 10.....	247

10.1 Anhang U1

10.1.1 Stichprobe U1

Nr.	Titel	Verlag / Herausgeber	Land	Art	Quelle
L1	Connected 3	LMVZ	CH	L	Fischer, C., & Meier, U. (2020). <i>Connected 3</i> (4. Auflage). Lehrmittelverlag Zürich.
L2	Einfach Informatik 7-9: Programmieren	Klett und Balmer Verlag	CH	L	Hromkovič, J., & Kohn, T. (2018). <i>Einfach Informatik 7-9. Programmieren: Schulbuch</i> (1. Auflage.). Baar Klett und Balmer Verlag 2018-. Hromkovič, J., & Kohn, T. (2018). <i>Einfach Informatik 7-9. Programmieren: Begleitband</i> (1. Auflage). Klett und Balmer Verlag.
L3	Gruppenpuzzle	PHSZ	CH	L	Hielscher, M., & Döbeli Honegger, B. (2019, September 23). <i>Scratch Gruppenpuzzle</i> . Pädagogische Hochschule Schwyz. https://ilearnit.ch/download/ScratchGruppenpuzzle.pdf
L4	PaperPenny	PHBern	CH	L	Pädagogische Hochschule Bern. (2024, März 18). <i>Paper Penny</i> . Paper Penny. https://paperpenny3.weebly.com/
L5	Vernetzt -digitale Grundbildung	öbv	Ö	L	Fikisz, W., Bucher, K., Ciprina, H. J., Meyer, M., Rottinger, U., & Schmid, J. (2023). <i>vernetzt—Digitale Grundbildung</i> (1. Auflage). Österreichischer Bundesverlag Schulbuch. https://www.oebv.at/flippingbook/9783209128294/II/
L6	Imperative Programmiersprache – Algorithmisches Problemlösen mit Kara	inf-schule	D	L	Becker, K., & Rosemann, F. (2024, März 18). <i>Algorithmisches Problemlösen mit Kara</i> . INF-Schule. https://www.inf-schule.de/imperative-programmierung/kara

L7	Imperative Programmiersprache – Algorithmisches Problemlösen mit Scratch	inf-schule	D	L	Becker, K., & Jochum, H. (2024, März 18). <i>Algorithmisches Problemlösen mit Scratch</i> . INF-Schule. Imperative Programmiersprache - Algorithmisches Problemlösen mit Scratch
L8	Einfach Programmieren	Carlsen	D	L	Knodel, D., & Knodel, P. (2023). <i>Einfach programmieren. Mit Arbeitsheft und App Programmieren lernen</i> (1. Auflage). Ernst Klett Verlag.
L9	Duden Informatik SEKI Baden -Württemberg	Cornelsen	D	L	Engelmann, L., Forman, F. X., Diethelm, I., & Breig, T. (2017). <i>Informatik—Sekundarstufe I - Baden Württemberg</i> . Duden Paetec Schulbuchverlag.
L10	Informatik 56 Nordrhein-Westfalen	Cornelsen	D	L	Asschoff, A., Brands, S., Cürlis, N., Kneblewski, M., Nattermann, F., Salloch, S., Watzlawek, K.-P., & Wellesen, A. (2021). <i>Informatik Gymnasium. 5/6, Schülerband</i> (Nordrhein-Westfalen). C.C. Buchner.
F11	Variable Evaluation: an Exploration of Novice Programmers' Understanding and Common Misconceptions	(Kohn, 2017)	D	F	Kohn, T. (2017). Variable Evaluation: An Exploration of Novice Programmers' Understanding and Common Misconceptions. <i>Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education</i> , 345–350. https://doi.org/10.1145/3017680.3017724
M12	Act 2 Curriculum	ECforAll	A	M	L: ECforALL. (2023). <i>Act 2 Curriculum</i> . ECforALL Curriculum. https://www.curriculum.elementarycomputingforall.org/act-2 <i>Material durch diverse Studien erprobt, aber nicht spezifisch für Variablen, deshalb keinen Forschungsbeitrag</i>
M13	Robot Boxes	Everyday Computing	A	M	F: Everyday Computing. (o. J.). <i>4th Grade Lesson: Introducing Variables: Robot Boxes</i> . Everyday Computing. http://everydaycomputing.org/lessons/action-fractions/grade-4/robot-boxes F: Rich, K. M., Franklin, D., Strickland, C., Isaacs, A., & Eathing, D. (2020). A Learning Trajectory for Variables Based in Computational Thinking Literature: Using Levels of Thinking

					to Develop Instruction. <i>Computer Science Education</i> , 32(2), 213–234. https://doi.org/10.1080/08993408.2020.1866938
F14	Variables	(Bagge & Grover, 2020)	A	F	Bagge, P., & Grover, S. (2020). Variables. In S. Grover (Hrsg.), <i>Computer Science in K-12: An A to Z handbook on teaching programming</i> (S. 227–239). Edfinity.
M15	Concepts before coding: Non-programming interactives to advance learning of introductory programming concepts in middle school.	(Grover et al., 2019)	A	M	L+F: Grover, S., Jackiw, N., & Lundh, P. (2019). Concepts before coding: Non-programming interactives to advance learning of introductory programming concepts in middle school. <i>Computer Science Education</i> , 29(2–3), 106–135. https://doi.org/10.1080/08993408.2019.1568955
F16	Introducing the Computer Science Concept of Variables in Middle School Science Classrooms	(Buffum et al., 2018)	A	F	L+F: Buffum, P. S., Ying, K. M., Zheng, X., Boyer, K. E., Wiebe, E. N., Mott, B. W., Blackburn, D. C., & Lester, J. C. (2018). Introducing the Computer Science Concept of Variables in Middle School Science Classrooms. <i>Proceedings of the 49th ACM Technical Symposium on Computer Science Education</i> , 906–911. https://doi.org/10.1145/3159450.3159545
F17	Thinking out of the box	(Hermans et al., 2018)	N	F	Hermans, F., Swidan, A., Aivaloglou, E., & Smit, M. (2018). Thinking out of the box: Comparing metaphors for variables in programming education. <i>Proceedings of the 13th Workshop in Primary and Secondary Computing Education</i> , 1–8. https://doi.org/10.1145/3265757.3265765

10.1.2 Kriterienraster U1

Kriterium	Item	Ausprägungen	Hinweise
Name, Verlag			
Art	1	Lehrmittel / Unterrichtsmaterial	Wenn vorhanden auch Begleitband / Kommentar für Lehrpersonen
	2	Forschungsbeitrag	
	3	Mischform	Durch einen Forschungsbeitrag entstandenes Unterrichtsmaterial / Lehrmittel
Land	1	Schweiz	Bei Deutschland und Österreich ist die Unterscheidung nicht immer ganz klar.
	2	Deutschland	
	3	Österreich	
	4	Amerika	
	5	Niederlande	
Zielstufe	5	5. Klasse	
	6	6. Klasse	
	7	7. Klasse	
	0	Unklar	
Vorangegangene Programmierkonzepte Auswertung für jedes Konzept einzeln	0/1	Sequenz	
	0/1	Schleife	
	0/1	Bedingung	
	0/1	Programme / Unterprogramme	
	0/1	Parameter	
	0/1	Boolean	
	f	Fehlende Werte	
			Keine Auswertung möglich
Analogie	1	Box	

Visuelle oder schriftliche Hinweise gelten	2	Lable	Wenn im Begleitband oder Kommentar für Lehrpersonen etwas über den Einsatz der Analogie gesagt wurde, dann wurden stichwortartige Kommentare dazu verfasst.
	3	Tabelle	
	4	Whiteboard	Resultate des Forschungsbeitrags wurden auch stichwortartig vermerkt.
	5	Briefumschlag	
	6	Alltagsanalogie	
	0	Keine	
Rolle von Variablen nach Sajaniemi (2006) Das einmalige Vorkommen wird ausgewertet. Mehrfaches Vorkommen wird nicht angeschaut.	0/1	Fixed Value	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den Wert nach der Initialisierung nicht mehr verändert. z.B. Benutzername in Spiel
	0/1	Stepper	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable eine vorher-sagbare Folge von Werten durchläuft. Der Abstand zwischen den aufeinander-folgenden Werten ist meistens konstant. z.B. Timer / Stoppuhr
	0/1	Most-Recent Holder	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den letzten Wert enthält, der beim Durchlaufen einer Folge von Werten gefunden wurde. z.B. Antworten auf Quizfragen
	0/1	One-way flag	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable nur zwei Werte annehmen kann (true / false) z.B. Korrektur von Antworten (richtig / falsch)
	0/1	Most-wandet Holder	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den bisher besten Wert enthält. Der aktuelle Wert ist also besser als alle vorherigen Werte. z.B. Rekord in einem Spiel
	0/1	Gatherer	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable, Werte akku-muliert. Der neue Wert ergibt sich aus der Kombination mit dem vorherigen Wert. z.B. Punktestand
	0/1	Follower	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable Werte ein-nimmt, welche von einer anderen Variable abhängen. Der Wert wird meist direkt nach der Aktualisierung für die Aktualisierung einer anderen Variable verwendet.

			z.B. Variable, die Benutzereingabe verarbeitet
	0/1	Temporary	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den Wert nur für eine sehr kurze Zeit enthält. z.B. Wertetausch
	f	Fehlende Werte	Keine Auswertung möglich
Themebereich nach Zhang und Nuri (2019) + Mathematik	1	Game	
	2	Storytelling	
	3	Musik	
	4	Mathematik	
	5	Animation	
	0	Nicht erkennbar	
Programmierlernumgebung	1	Scratch	
	2	Python / TigerJython	
	3	Blockly	
	4	eigenes	Es wurde extra für eine Programmierlernumgebung entwickelt
	0	Keine	
Übungscharakter		Plugged	Es wurden nur Aktivitäten gefunden, welche mit einem Gerät gelöst werden können.
		Unplugged	Es wurden nur Aktivitäten gefunden, welche ohne Geräte gelöst werden können.
		Beides	Beide Arten von Aufgaben wurden gefunden.

10.1.3 Auswertung U1

Die Auswertung wurde in einem Excel-Dokument gemacht und liegt dieser Masterarbeit separat bei.

10.1.3 Auswertung U1

Die Auswertung wurde in einem Excel-Dokument gemacht und liegt dieser Masterarbeit separat bei.

10.2 Anhang U2

10.2.1 Anfragetext U2

Liebe / Lieber (bitte anpassen)

Zurzeit schreibe ich meine Masterarbeit zum Thema "Informatikunterricht im Zyklus 2: Variablen". Dabei interessiere ich mich für Aktivitäten und Analogien, die das Verständnis und das effektive Benutzen von Lernenden im Alter von 10-12 Jahren fördern. Im Rahmen dieses Forschungsthemas entwickle ich eine prototypische Unterrichtsidee, die ich gerne mit einer Fachperson auf diesem Gebiet besprechen möchte. Das Interview ist wie folgt strukturiert:

- Vorbereitung: Lesen des Prototyps (Dauer ca. 30 min)
- Interviewsetting: Online-Interview über Zoom
- Dauer: 45 min
- Fragen zum Programmierunterricht für Lernende in diesem Alter und wie dieser aufgebaut sein muss, um das Verständnis von Variablen zu fördern.
- Diskussion der Unterrichtseinheit
- Das Interview wird aufgezeichnet.
- Zeitrahmen: Das Interview findet Ende November/Anfang Dezember statt.

Hast du Interesse und Zeit an diesem Interview teilzunehmen? Ich würde mich sehr über deine Teilnahme und den fachlichen Austausch freuen.

Vielen Dank im Voraus und herzliche Grüsse

Lea Gisler

Dear / Dear (please adapt)

I am currently writing my Master's thesis on the topic "Computer Science Teaching in Cycle 2: Variables". In doing so, I am interested in activities and analogies that promote understanding and effective use for learners aged 10-12. As part of this research topic, I am developing a prototypical teaching idea that I would like to discuss with an expert in the field. The interview is structured as follows:

- Preparation: Reading of the prototype (duration approx. 30 min).
- Interview setting: online interview via Zoom
- Duration: 45 min
- Questions about programming lessons for learners of this age and how they need to be structured to promote understanding of variables.
- Discussion of the lesson
- The interview will be recorded.
- Time frame: The interview will take place in late November/early December.

Are you interested and have time to participate in this interview? I would be very happy about your participation and the professional exchange.

Many thanks in advance and best regards

Lea Gisler

10.2.2 Einverständniserklärung deutsch U2

Einverständniserklärung zur Teilnahme am Interview «Informatikunterricht im Zyklus 2: Variablen»

Altdorf, 04.12.2023

Im Rahmen der Masterarbeit «Informatikunterricht im Zyklus 2: Variablen» werden Expert:innen zum Thema Variablen im Programmierunterricht befragt. Das Interview (Dauer ca. 45 min) findet online statt und wird aufgezeichnet. Die Daten werden ausgewertet und fliessen in die Erarbeitung eines Prototyps ein. Nach Abschluss der Masterarbeit (voraussichtlich November 2024) werden die Aufzeichnungen gelöscht.

Die Ergebnisse der gesamten Studie werden in der schriftlichen Masterarbeit veröffentlicht, im Rahmen des Studiengangs präsentiert und eventuell später in einer wissenschaftlichen Zeitschrift publiziert oder auf einer Konferenz präsentiert. Auf Wunsch kann die Arbeit nach der Veröffentlichung eingesehen werden.

Expert:in:

Datum des Interviews:

Mit meiner Unterschrift erkläre ich mich unter den oben genannten Bedingungen zur Teilnahme am Interview bereit und bin damit einverstanden, dass die aufgezeichneten Daten zu Forschungszwecken verwendet werden.

Datum / Unterschrift

Herzlichen Dank für das Engagement!

Lea Gisler, lea.gisler@stud.phsz.ch
Studentin PHSZ Fachdidaktik Medien & Informatik

Pädagogische
Hochschule Schwyz
Lea Gisler & Prof. Dr. Eva Marinus
Forschungsprofessur Fachdidaktik Medien und Informatik
eva.marinus@phsz.ch,
www.phsz.ch

10.2.3 Einverständniserklärung englisch U2

Declaration of consent to participate in the interview

"Computer science teaching in cycle 2: variables"

Altdorf, 04.12.2023

As part of the master's thesis "Computer Science Teaching in Cycle 2: Variables", experts are interviewed on the topic of variables in programming lessons and on a prototype that has been created. The interview (duration approx. 45 min) takes place online and is recorded. The data will be evaluated and will flow into the development of a prototype. After completion of the master thesis (expected November 2024), the recordings will be deleted.

The results of the entire study will be published in the written master's thesis, presented as part of the course, and possibly later published in a scientific journal or presented at a conference. Upon request, the thesis will be available for review after publication.

Expert:

Date of the interview:

By signing below, I agree to participate in the interview under the above conditions and consent to the recorded data being used for research purposes.

Date / Signature

Many thanks for the commitment!

Lea Gisler, lea.gisler@stud.phsz.ch
Student PHSZ Didactics Media & Computer Science

Pädagogische Hochschule
Schwyz
Lea Gisler & Prof. Dr. Eva
Marinus
Forschungsprofessur Fachdidaktik Medien und Informatik
eva.marinus@phsz.ch
www.phsz.ch

10.2.4 Interviewleitfaden deutsch U2

Einleitung

Herzlichen Dank für deine Teilnahme an Interview. Im Rahmen meiner Masterarbeit an der PH Schwyz interessiere ich mich für den Programmierunterricht von Lernenden der 5.-6. Klasse (Lernenden im Alter von 10-12 Jahren). Ich interessiere mich dabei für das Programmierkonzept «Variable» und wie dieses für Lernende in diesem Alter erklärbar gemacht werden kann. Ich interessiere mich dafür für Analogien und Aktivitäten, die Kindern helfen das Konzept zu verstehen. Das Transkript und die Aufnahme werden nur im Rahmen dieser Masterarbeit verwendet. Nach Beendigung des Projekts wird die Aufnahme gelöscht. Hast du dazu noch eine Frage?

PPT zeigen

Zu Beginn des Interviews sind die Fragen eher offen gestalten, am Ende möchte ich genauer auf gewisse Aspekte eingehen. Ich gehe dabei auf die Themen Kinder und Variablen, Informatikunterricht mit Variablen und Analogien für Variablen ein.

Kinder und Variablen

Welche Erfahrungen hast du selbst gemacht beim Programmieren mit Variablen?

Könntest du Erfahrungen und Beobachtungen beschreiben, wie Kinder deiner Meinung nach mit Variablen umgehen?

Welche Erfahrungen hast du gemacht, wie Lernende Variablen beschreiben?

Welches Verständnis von Variablen hältst du für angemessen für Kinder im Alter von 10-12 Jahren?

Informatikunterricht mit Variablen

Mich würde interessieren, wie für dich guter Programmierunterricht aussieht?

Wie würdest du das Programmierkonzept «Variablen» Lernenden im Alter von 10-12 Jahren mit wenig Programmiererfahrung erklären?

Vielleicht nach oben Welche Aktivitäten sind wichtig, um das Verständnis von Variablen zu fördern?

- Check: Unplugged, plugged, Programmierumgebung,

Welches Vorwissen brauchen Kinder, um das Konzept der Variablen zu verstehen?

evtl. Welche Konzepte sollten Lernende verstehen, bevor sie sich mit Variablen beschäftigen?

- Check: Programmierkonzepte Schleifen, Bedingungen, Parameter, Programme

Welche Herausforderungen haben Kinder beim Verstehen von Variablen deiner Meinung nach?

Analogien für Variablen

Welche Analogie für Variablen würdest du in deinem Unterricht verwenden, wie würdest du diese einsetzen, auf was würdest du achten?

Welche weiteren Analogien kennst du, welche für Variablen verwendet werden?

Wie schätzt du die Verwendung von Analogien für Variablen im Informatikunterricht ein? Welche Chancen und Risiken gibt es da?

Als Nächstes möchte ich ein Brainstorming machen, wie die Verwendung bestimmter Analogien zur Erklärung von Variablen zu bestimmten Missverständnissen führen könnte. In dieser Tabelle sind häufige Fehlvorstellungen zu erkennen. Du kannst diese gerne kurz durchlesen.

Schauen wir uns zuerst die Analogie Box mit Post-Its an. Siehst du da einen Zusammenhang zwischen dieser Analogie und den Fehlvorstellungen?

Wenn jetzt in die Box Kugeln entsprechend dem Wert der Variable sind, verändert das etwas in Bezug auf die Fehlvorstellungen?

Je nach Antworten von oben: genannte Analogien einbauen, sonst mit eigenen Analogien aus der Liste weiterfahren.

Abschlussfrage:

Gibt es noch etwas, was du zu diesem Thema ergänzen möchtest?

Ich danke dir für deine Zeit, deine Gedanken zum Thema. Dieser fachliche Austausch hat mir für meine Masterarbeit sehr geholfen. Wenn du Interesse hast, kann ich dir gerne nach Abschluss meiner Masterarbeit die Ergebnisse zukommen lassen.

10.2.5 Interviewleitfaden englisch U2

Introduction

Thank you very much for your participation in the interview. As part of my master's thesis at the PH Schwyz, I am interested in teaching programming to 5th-6th grade learners (learners aged 10-12 years). I am interested in the programming concept "variable" and how it can be explained to learners of this age. So I am looking closer at activities and analogies, visualizations and explanations that help them understand this concept. The transcript and the recording will only be used for the purpose of this master thesis. After completing the project, the recording will be deleted. Do you have any questions?

Show Miro-Board

At the beginning of the interview, the questions are more openly formulated, but at the end I will ask you more specific questions. I will cover the topics of children and variables, teaching computer science with variables and analogies for teaching variables.

Children and variables

Could you please share some experiences and observations of how you have seen children deal with variables?

and how, in your experience do they typically describe variables?

What level of understanding of variables do you think is appropriate for children aged 10-12/ Grades 5/6?

What challenges do you think children have in understanding variables?

Teaching computer science with variables

How would you explain the programming concept of "variables" to learners aged 10-12 with little programming experience?

What activities are important to promote understanding of variables?

- Check: Unplugged, plugged, programming environment,

What prior knowledge do children need to understand the concept of variables?

Possibly: What concepts should learners understand before dealing with variables?

- Check: Programming concepts loops, conditions, parameters, programs

What challenges do you think teachers have when they teach children about variables?

Analogies for variables

Which analogy for variables would you use in your lessons?

How would you use it?

What would you pay attention to?

What other analogies do you know that are used for variables?

What do you think of using of analogies for variables in computer science lessons? What opportunities and risks are there?

Next, I would like to brainstorm with you about how using specific analogies for explaining variables could lead to specific misconceptions or in a positive way kind of prevent them.

This table shows common misconceptions. Feel free to read through them briefly.

First, let's look at the analogy box with post-its. Do you see a connection between this analogy and the misconceptions?

If there are now pebbles in the box corresponding to the value of the variable, does this change anything in relation to the misconceptions?

Depending on the answers from above: incorporate the analogies mentioned, otherwise continue with my own analogies from the list.

Final question

Is there anything else you would like to add on this topic?

Thank you for your time and your thoughts on the topic. This professional exchange was very helpful for my Master's thesis. If you are interested, I would be happy to send you the results once I have completed my Master's thesis.

10.2.6 Präsentation Interview U2



Procedure

Children and variables

Teaching computer science with variables

Analogies for variables in computer science

Brainstorming analogies and misconceptions

Expert:inneninterview

2

Procedure

Children and variables

Teaching computer science with variables

Analogies for variables in computer science

Brainstorming analogies and misconceptions

Expert:inneninterview

3

Procedure

Children and variables

Teaching computer science with variables

Analogy for variables in computer science

Brainstorming analogies and misconceptions

Expert:inneninterview

4

Procedure

Children and variables

Teaching computer science with variables

Analogy for variables in computer science

Brainstorming analogies and misconceptions

Expert:inneninterview

5

Misconceptions for variables

Misconceptions
A variable can hold multiple values at a time / 'remembers' old values.
Students believe that the variable "a" will contain the first assigned value.
Students believe that the value assigned to the variable "a" will be the sum of every previously assigned value.
A variable is (merely) a pairing of a name to a changeable value (with a type). It is not stored inside the computer.
Primitive assignment stores equations or unresolved expressions.
Assignment moves a value from a variable to another.
Incrementing a counter variable is an indivisible operation (no separate evaluation of right-hand side).
Primitive assignment works in opposite direction.
Primitive assignment works both directions (swaps).

Expert:inneninterview

6

Misconceptions for variables

Misconceptions
A variable can hold multiple values at a time / 'remembers' old values.
Students believe that the variable "a" will contain the first assigned value.
Students believe that the value assigned to the variable "a" will be the sum of every previously assigned value.
A variable is (merely) a pairing of a name to a changeable value (with a type). It is not stored inside the computer.
Primitive assignment stores equations or unresolved expressions.
Assignment moves a value from a variable to another.
Incrementing a counter variable is an indivisible operation (no separate evaluation of right-hand side).
Primitive assignment works in opposite direction.
Primitive assignment works both directions (swaps).

Expert:inneninterview

Variablen



7

Misconceptions for variables

Misconceptions
A variable can hold multiple values at a time / 'remembers' old values.
Students believe that the variable "a" will contain the first assigned value.
Students believe that the value assigned to the variable "a" will be the sum of every previously assigned value.
A variable is (merely) a pairing of a name to a changeable value (with a type). It is not stored inside the computer.
Primitive assignment stores equations or unresolved expressions.
Assignment moves a value from a variable to another.
Incrementing a counter variable is an indivisible operation (no separate evaluation of right-hand side).
Primitive assignment works in opposite direction.
Primitive assignment works both directions (swaps).

Expert:inneninterview

Variablen



8

Misconceptions for variables

Misconceptions
A variable can hold multiple values at a time / 'remembers' old values.
Students believe that the variable "a" will contain the first assigned value.
Students believe that the value assigned to the variable "a" will be the sum of every previously assigned value.
A variable is (merely) a pairing of a name to a changeable value (with a type). It is not stored inside the computer.
Primitive assignment stores equations or unresolved expressions.
Assignment moves a value from a variable to another.
Incrementing a counter variable is an indivisible operation (no separate evaluation of right-hand side).
Primitive assignment works in opposite direction.
Primitive assignment works both directions (swaps).

Expert:inneninterview

Variablen

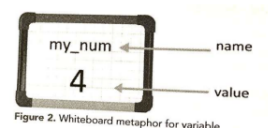


Figure 2. Whiteboard metaphor for variable

9

10.2.7 Kodierleitfaden U2

Hauptkategorie / Unterkategorie	Definition	Ankerbeispiel
Analogien Diese Kategorie wird nur im ersten Teil codiert, da im zweiten Teil explizit Analogien vorgestellt werden.		
Box-Analogie	Es wird die Verwendung der Boxanalogie zur Erklärung von Variablen beschrieben. In die Box werden Zettel gelegt, welche den Wert darstellen.	Ich würde mir da wirklich irgendein Kuvert, ein Böxli oder eine kleine Kiste, wo dieser Wert drin ist. So etwas könnte man. (...) Wie so ein Platzhalter. (Expertin 1, Pos. 40)
Box-Analogie mit Gegenständen	Es wird die Verwendung der Boxanalogie zur Erklärung von Variablen beschrieben. In die Box werden Gegenstände wie Murmeln oder Punkte gelegt, welche den Wert darstellen.	Ich glaube, ich würde wirklich ein Zündholzschachtel mitnehmen. (..) Dann würden wir sagen, wir rechnen jetzt damit irgendwie so. Dann könnte man es ja auch mit einem Flussdiagramm durchschicken. Dann würden wir diesen Wert wirklich immer wieder verändern. Vielleicht dann mit diesen Zählpunkten könnte man da Perlen oder irgendetwas könnte man fünf reinton. (Expertin 3, Pos. 42)
Label-Analogie	Es wird die Verwendung eines Labels zur Erklärung von Variablen beschrieben. Ein Label kann einem Wert zugeordnet werden. Es kommt in diese Kategorie, wenn das Aussehen des Labels nicht genauer erklärt wird.	And so, that led us to using flat boxes, like a square or a rectangle, and using post-its, right? One post-it at a time. (Expertin 2, Pos. 58)
Whiteboard-Analogie	Es wird die Verwendung der Whitboardanalogie zur Erklärung von Variablen beschrieben. Der Name sowie der Wert werden auf das Whiteboard geschrieben.	So, some other things that were suggested were, like using a name tag analogy, like maybe, or like a slate or a whiteboard, where you write something and you wipe it away and to write something else. (Expertin 2, Pos. 58)
Alltagsanalogie	Es wird die Verwendung einer Alltagsvariable zur Erklärung von Variablen beschrieben. Es wird ein explizites Beispiel aus dem Alltag genannt, dass Eigenschaften von Variablen verdeutlichen soll.	Yes. But yeah, I think connecting to anything that kids, you know, even a digital clock, what time is it? Actually, I never thought of that before. Like we can think about the idea that there is a time. Yeah, it's going to be different at different points. (Expertin 1, Pos. 28)

Einschätzungen zu den Analogien			
Box-Analogie	Stärken	Es werden positive Aspekte der Box-Analogie beschrieben. Aspekte, welche von dieser Analogie besonders gut verdeutlicht werden oder welches Verständnis diese Analogie fördern kann.	So even just thinking there's something in the box, I can put whatever I want in the box, take it out, put something else in. (.) That's really useful, I think, for the concept, again, having that mental model for a thing that you can put something into, but you don't know what it is yet, the box is closed. So like that's if that's what I'm trying to get to is that basic idea of an unknown, then the box is useful (Expertin 1, Pos. 30)
	Schwächen	Es werden negative Aspekte der Box-Analogie beschrieben. Aspekte, welche diese Analogie nicht gut verdeutlicht werden oder Fehlvorstellungen, die durch diese Analogie gefördert werden können.	So, can you see that this is a, like a shallow box here? So, if I say this is my variable, I'm going to put something into it, then they can understand that this, this variable could be empty or it could be full with something. But oftentimes, we have kids, you could put more and more things in here. And so, that indicates that there would be a misconception. They would think all the stuff you put in can still be in there. And so, there was a tension between like this container and its implied three-dimensionality. (Expertin 2, Pos. 58)
Box-Analogie mit Gegenständen	Stärken	Es werden positive Aspekte der Box-Analogie mit Gegenständen beschrieben. Aspekte, welche von dieser Analogie besonders gut verdeutlicht werden oder welches Verständnis diese Analogie fördern kann.	// Als Ankerbeispiel gilt das von der Box-Analogie.
	Schwächen	Es werden negative Aspekte der Box-Analogie mit Gegenständen beschrieben. Aspekte, welche diese Analogie nicht gut verdeutlicht werden oder Fehlvorstellungen, die durch diese Analogie gefördert werden können.	// Als Ankerbeispiel gilt das von der Box-Analogie
Label-Analogie	Stärken	Es werden positive Aspekte der Label-Analogie beschrieben. Aspekte, welche von dieser Analogie besonders gut verdeutlicht werden oder welches Verständnis diese Analogie fördern kann.	// Als Ankerbeispiel gilt das von der Box-Analogie.
	Schwächen	Es werden negative Aspekte der Label-Analogie beschrieben. Aspekte, welche diese Analogie nicht gut verdeutlicht werden oder Fehlvorstellungen, die durch diese Analogie gefördert werden können.	// Als Ankerbeispiel gilt das von der Box-Analogie
Whiteboard-Analogie	Stärken	Es werden positive Aspekte der Whiteboard-Analogie beschrieben. Aspekte, welche von dieser Analogie besonders gut verdeutlicht werden oder welches Verständnis diese Analogie fördern kann.	// Als Ankerbeispiel gilt das von der Box-Analogie.
	Schwächen	Es werden negative Aspekte der Whiteboard-Analogie beschrieben. Aspekte, welche diese Analogie nicht gut verdeutlicht werden oder Fehlvorstellungen, die durch diese Analogie gefördert werden können.	// Als Ankerbeispiel gilt das von der Box-Analogie

Briefumschlag- Analogie	Stärken	Es werden positive Aspekte der Briefumschlags-Analogie beschrieben. Aspekte, welche von dieser Analogie besonders gut verdeutlicht werden oder welches Verständnis diese Analogie fördern kann.	// Als Ankerbeispiel gilt das von der Box-Analogie.
	Schwächen	Es werden negative Aspekte der Briefumschlags-Analogie beschrieben. Aspekte, welche diese Analogie nicht gut verdeutlicht werden oder Fehlvorstellungen, die durch diese Analogie gefördert werden können.	// Als Ankerbeispiel gilt das von der Box-Analogie
Alltagsanalogie	Stärken	Es werden positive Aspekte der Alltagsanalogie beschrieben. Aspekte, welche von dieser Analogie besonders gut verdeutlicht werden oder welches Verständnis diese Analogie fördern kann.	// Als Ankerbeispiel gilt das von der Box-Analogie.
	Schwächen	Es werden negative Aspekte der Alltagsanalogie beschrieben. Aspekte, welche diese Analogie nicht gut verdeutlicht werden oder Fehlvorstellungen, die durch diese Analogie gefördert werden können.	// Als Ankerbeispiel gilt das von der Box-Analogie
Aktivitäten / Aufgaben zum Thema Variablen			
Unplugged	Es wird eine Aktivität beschrieben, die ohne Geräte gemacht werden kann. Es wird die gesamte Aktivität codiert		So sometimes we do things like input output machines. We talk about that. So I put in a five, I get out a six. What's happening in that? (Expertin 1, Pos. 18)
Plugged	Es wird eine Aktivität beschrieben, für welche die L ein Gerät brauchen. Es wird die gesamte Aktivität codiert.		Und nachher vielleicht auch sogar ein Programm geben, bei dem sie eine Variable einfügen müssen, damit sie wie ein fixes, fertiges Programm haben und dann sehen, wie kann ich jetzt dieses Programm halt eben wirklich optimieren oder wo macht es Sinn, eine Variable einzufügen? (Expertin 3, Pos. 26)
Rollen von Variablen			
Bei den beschriebenen Aktivitäten wird die Rolle der darin vorkommenden Variablen kategorisiert.			
Fixed Value	Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den Wert nach der Initialisierung nicht mehr verändert.		Dann würde ich aber auch versuchen, nachher eine Übung machen, wo nicht konkret dieser Wert immer Thema ist, sondern wo wir etwas anderes rechnen und sagen wollen, jetzt rechnen wir das mal dieser Wert. (.) Das ist dann die Klasse zum Beispiel, da haben wir eine zweite Zündholzschachtel, wie viele Klassenkinder haben wir. Dann rechnen wir alles durch, vielleicht ein Apéro vorbereiten für irgendeinen Abschluss. (Expertin 3, Pos. 42)

Stepper		Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable eine vorher-sagbare Folge von Werten durchläuft. Der Abstand zwischen den aufeinander-folgenden Werten ist meistens konstant.	// keines Vorhanden
Most-recent holder		Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den letzten Wert enthält, der beim Durchlaufen einer Folge von Werten gefunden wurde.	And every verse is the same except for the value that's changing. (.) And so, kids get it. Right? And so, you could see kids coding, (.) you know, writing a computer program that will cycle through a familiar, something that's familiar to them from their childhood (Expertin 2, Pos. 34)
Most-wanted holder		Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den bisher besten Wert enthält. Der aktuelle Wert ist also besser als alle vorherigen Werte.	// keines Vorhanden
Gatherer		Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable, Werte ak-kumuliert. Der neue Wert ergibt sich aus der Kombination mit dem vorherigen Wert.	Ich weiss, dass wir vorlängerer Zeit mal ein Spiel programmiert haben. (.) Da ging es darum, auch Punkte zu sammeln. (Expertin 3, Pos. 6)
Follower		Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable Werte ein-nimmt, welche von einer anderen Variable abhängen. Der Wert wird meist di-rekt nach der Aktualisierung für die Aktualisierung einer anderen Variable ver-wendet.	// keines Vorhanden
Temporary		Es wird eine Aktivität / Aufgabe beschrieben, bei der eine Variable den Wert nur für eine sehr kurze Zeit enthält.	// keines Vorhanden
Didaktische Hinweise			
Vorstellungen über Variablen	Fehlvorstellungen über Variablen	Es wird eine Fehlvorstellung beschrieben, welche L im Alter von 10-12 Jahren über Variablen im Bereich der Informatik haben können.	But oftentimes, we have kids, you could put more and more things in here. And so, that indicates that there would be a misconception. They would think all the stuff you put in can still be in there (Expertin 2, Pos. 58)
	Vorstellungen	Es wird beschrieben welche Vorstellungen L im Alter von 10-12 Jahren über Variablen im Bereich der Informatik haben können.	So I think they can understand that I've been able to get them to think about a variable as something that can change. (Expertin 1, Pos. 14)

	Herausforderungen	Es wird beschrieben, welche Schwierigkeiten L im Alter von 10-12 Jahren haben beim Lösen von Aufgaben oder Erstellen von Programmen mit Variablen.	But we've only tried in one lesson, at least in my work, to really get them to. (..) Sort of replace a static number with a variable, like integrate the variable into the code, and they were able to do that, but what they really struggled with was there has to be a place where you are setting the first value of the variable. So it's that I think initialization is what I'm trying to get at. They are able to integrate variables into code, but they don't. (..) They tend to miss the step where they have to tell the computer at least what the initial value of the variable is. (Expertin 1, Pos. 16)
Didaktische Hinweise für Variablen		Es werden Hinweise zum Vermitteln des Programmierkonzepts Variable gegeben.	The other thing I would say to you is that. I like to, you know, I like to talk to children about the meaning of a word sometimes before I tell them what the word is. Right. And so kids know the word change and they know in English, they know the word change. They know the word value, but they don't. And even though variable has vary at the beginning of it, vary is not a word that young children tend to use. Right. And so their understanding of variable, their understanding of any concept is very much tied to language that is familiar to them. And so when we do in our lesson that introduces variables, we do not say the word variable until the wrap up. (Expertin 2, Pos. 96)
Analogie als didaktisches Konzept	Schwächen	Es wird beschrieben, wieso und unter welchen Bedingungen Analogien helfen das Programmierkonzept Variable besser zu verstehen.	And so you can see I'm struggling with the words for even right now. So I think finding ways to talk about it to students is really hard, which I guess kind of relates to the idea of analogies, which we'll talk about, too. (Transkript KR, Pos. 26)
	Stärken	Es wird beschrieben, welche Schwierigkeiten oder auch Risiken sich bei der Verwendung von Analogien zur Erklärung des Programmierkonzepts Variablen ergeben können.	No analogy is perfect, right? So it depends on the part of the the concept you're trying to really hit home. (Transkript KR, Pos. 30)

10.2.8 – 10.2.10 Interviewtranskript 1-3

Für Veröffentlichung entfernt.

10.3 Anhang U3

10.3.1 Einverständniserklärung U3

pädagogische hochschule schwyz

Einverständniserklärung zur Teilnahme am Interview «Informatikunterricht im Zyklus 2»

Altdorf, 22.04.2024

Sehr geehrte Eltern

Im Modullehrplan «Medien und Informatik» sind unter anderem Kompetenzen zum Bereich Programmieren formuliert. Kinder sollen dabei verschiedene Programmierkonzepte wie die Schleife oder Bedingung kennenlernen. Im Rahmen meiner Masterarbeit an der Pädagogischen Hochschule Schwyz untersuche ich deshalb, wie die Einführung solcher Konzepte am besten gestaltet werden kann.

Um die Vorstellungen von Schülerinnen und Schülern genauer zu verstehen, möchte ich mit Ihrem Kind ein Interview (Dauer ca. 30 min) durchführen. Das Interview findet während dem Unterricht in einem Gruppenraum statt und wird aufgezeichnet. Es wird ein Transkript, eine Verschriftlichung des Gesprächs, erstellt. Diese Daten werden anonymisiert ausgewertet, so dass keinerlei Rückschlüsse auf einzelne Lernende möglich sind. Die persönlichen Informationen (Name, Geburtsdatum, Geschlecht) werden sicher und getrennt von der Datenauswertung gespeichert. Alle persönlichen Angaben werden spätestens bis Ende des Projektes gelöscht (voraussichtlich November 2024).

Die Ergebnisse der gesamten Studie werden in der schriftlichen Masterarbeit veröffentlicht, im Rahmen des Studiengangs präsentiert und eventuell später in einer wissenschaftlichen Zeitschrift publiziert oder auf einer Konferenz präsentiert. Auf Wunsch kann die Arbeit nach der Veröffentlichung eingesehen werden.

Es würde mich sehr freuen, wenn Ihr Kind an diesem Interview teilnimmt. Bitte füllen Sie und Ihr Kind den Talon unten zur Einverständniserklärung aus und geben Sie diesen bis spätestens **Freitag, 26. April 2024** der Klassenlehrperson zurück. Besten Dank.

Bei allfälligen Fragen gebe ich gerne detailliertere Auskunft (lea.gisler@stud.phsz.ch)

Herzlichen Dank und freundliche Grüsse

Lea Gisler, lea.gisler@stud.phsz.ch
Studentin PHSZ Fachdidaktik Medien & Informatik

Pädagogische
Hochschule Schwyz
Lea Gisler & Prof. Dr. Eva Marinus
Forschungsprofessur Fachdidaktik Medien und Informatik
eva.marinus@phsz.ch,
www.phsz.ch

Einverständniserklärung zur Teilnahme am Interview «Informatikunterricht im Zyklus 2»

Die Teilnahme am Interview ist freiwillig. Auch wenn Sie Ihr Einverständnis zu einem späteren Zeitpunkt vor Ende des Projekts zurückziehen möchten, werden die Daten Ihres Kindes selbstverständlich gelöscht.

Name des Kindes: _____

Name der Lehrperson: _____

- Ja, unter den oben genannten Bedingungen erklären wir uns bereit, dass unser Kind am Interview teilnimmt. Wir sind damit einverstanden, dass das Gespräch aufgenommen, transkribiert und anonymisiert für Forschungszwecke verwendet wird.
- Nein, wir möchten nicht, dass unser Kind an diesem Interview teilnimmt.

Resultate der Studie

- Ja, ich möchte über die Resultate der Studie informiert werden (voraussichtlich im November 2024).

Falls ja: Bitte E-Mail-Adresse angeben:

Datum / Unterschrift des Kindes

Datum / Unterschrift der Eltern

10.3.2 Interviewleitfaden U3

Interviewleitfaden „Programmierkonzept Variable im Zyklus 2“

Vor Interview

- Demographische Angaben erfassen in Tabelle + Protokoll hervorheben
- Scratchprogramme starten / kontrollieren
- Aufnahme Starten (Handy + Laptop als Backup)
- Information Gespräch auf Hochdeutsch

Einleitung

Herzlichen Dank für deine Teilnahme am Interview. Ich werde das Gespräch aufnehmen, ist das in Ordnung für dich?

Ich werde dir anschliessend Fragen stellen, zu denen du ganz offen antworten kannst. Du hilfst mir damit, besser zu verstehen, was Kinder über das Programmieren wissen. Es gibt keine richtige oder falsche Antwort. Mich interessiert, was du denkst. Du bist für mich heute eine Expertin / ein Experte. Wenn ich etwas genauer wissen möchte, werde ich nachfragen. Ich werde dir auch verschiedene Programme in Scratch zeigen. Diese darfst du selbst ausprobieren. Während dieser Phase werde ich dich dazu auffordern laut zu denken. Du kannst mir dann sagen, was du siehst oder auch was du machst und wieso.

Aufnahme starten

Was machst du gerne in der Schule? Was machst du gerne am Computer? Was machst du gerne im Fach Medien & Informatik? Deine Lehrperson hat mir erzählt, dass ihr in der Schule auch schon programmiert habt. Kannst du dich noch erinnern, was ihr gemacht habt? Hast du in der Freizeit oder auch sonst weiter programmiert? Wenn ja, was hast du gemacht?

Alltagsvariable Temperatur

Leitfrage (Erzählaufforderung)	Subfragen / Nachfragen (auf Antworten reagieren)	Frage zur Begründung	Aufrechterhaltungs- und Steuerungsfragen	Material
Du kennst bestimmt den Begriff Temperatur. Erzähl mit alles, was du darüber weisst.	Was findest du spannend an der Temperatur? Welchen Einfluss hat die Temperatur auf deinen Alltag? Welche Eigenschaften hat Temperatur?	Warum? Kannst du ein Beispiel dafür machen?	Das ist sehr spannend, was du sonst noch? Kannst du mir das noch genauer erklären. Was hast du über die Temperatur in der Schule gelernt?	-
Du siehst nun ein Scratchprogramm auf dem Bildschirm. Klicke auf die grüne Flagge und lies die Anleitung durch. Hast du eine Frage dazu? Bediene nun das Programm und denke dabei laut. Kannst du bitte alles sagen, was dir durch den Kopf geht. Klicke auf die Leertaste, um das Programm zu starten.	Wie funktioniert das Programm? Was ist das? (auf Variable Temperatur zeigen) Was macht das? Was hat die Temperatur mit diesem Programm zu tun?	Da hast du etwas gezögert/gelacht – woran hast du gedacht? Wieso hast du das gedrückt/geklickt?	Was siehst du? Was fällt dir auf? Was machst du gerade? Das ging sehr schnell. Was ist dir beim Bedienen des Programms durch den Kopf gegangen?	Programm Temperatur
Du siehst nun ein Scratchspiel auf dem Bildschirm. Klicke auf die grüne Flagge und lies die Anleitung durch. Hast du eine Frage dazu? Bediene nun das Programm und denke dabei laut. Kannst du bitte alles sagen, was dir durch den Kopf geht. Klicke auf die Leertaste, um das Programm zu starten.	Wie funktioniert das Programm? Was ist das? (auf Variable Temperatur zeigen) Was macht das? Was ist das? (auf Variable Punktestand zeigen) Was macht das? Was hat die Temperatur mit diesem Programm zu tun?	Da hast du etwas gezögert/gelacht – woran hast du gedacht? Wieso hast du das gedrückt/geklickt?	Was siehst du? Was fällt dir auf? Was machst du gerade? Das ging sehr schnell. Was ist dir beim Bedienen des Programms durch den Kopf gegangen?	Programm Temperatur Spiel (Spiel ist auf 90 Sekunden eingestellt)

Alltagsvariable Zeit

Leitfrage (Erzählaufforderung)	Subfragen / Nachfragen (auf Antworten reagieren)	Frage zur Begründung	Aufrechterhaltungs- und Steuerungsfragen	Material
Du kennst bestimmt den Begriff Zeit. Erzähl mit alles, was du darüber weisst.	Was findest du spannend an der Zeit? Welchen Einfluss hat die Zeit auf deinen Alltag? Welche Eigenschaften hat Zeit?	Warum? Kannst du ein Beispiel dafür machen?	Das ist sehr spannend, was du sonst noch? Kannst du mir das noch genauer erklären. Was hast du über die Zeit in der Schule gelernt?	-
Du siehst nun ein Scratchprogramm auf dem Bildschirm. Klicke auf die grüne Flagge und lies die Anleitung durch. Hast du eine Frage dazu? Bediene nun das Programm und denke dabei laut. Kannst du bitte alles sagen, was dir durch den Kopf geht. Klicke auf die Leertaste, um das Programm zu starten.	Wie funktioniert das Programm? Was ist das? (auf Zeit zeigen) Was macht das? Was hat die Zeit mit diesem Programm zu tun?	Da hast du etwas gezögert/gelacht – woran hast du gedacht? Wieso hast du das gedrückt/geklickt?	Was siehst du? Was fällt dir auf? Was machst du gerade? Das ging sehr schnell. Was ist dir beim Bedienen des Programms durch den Kopf gegangen?	Programm Zeit Alltag
Du siehst nun ein Scratchspiel auf dem Bildschirm. Klicke auf die grüne Flagge und lies die Anleitung durch. Hast du eine Frage dazu? Bediene nun das Programm und denke dabei laut. Kannst du bitte alles sagen, was dir durch den Kopf geht. Klicke auf die Leertaste, um das Programm zu starten.	Wie funktioniert das Programm? Was ist das? (auf Variable zeigen) Was macht das? Was hat die Zeit mit diesem Programm zu tun?	Da hast du etwas gezögert/gelacht – woran hast du gedacht? Wieso hast du das gedrückt/geklickt?	Was siehst du? Was fällt dir auf? Was machst du gerade? Das ging sehr schnell. Was ist dir beim Bedienen des Programms durch den Kopf gegangen?	Programm Zeit Alltag Spiel Programm Zeit Spiel

Alltagsvariable Geschwindigkeit

Leitfrage (Erzählaufforderung)	Subfragen / Nachfragen (auf Antworten reagieren)	Frage zur Begründung	Aufrechterhaltungs- und Steuerungsfragen	Material
Du kennst bestimmt den Begriff Geschwindigkeit. Erzähl mit alles, was du darüber weisst.	Was findest du spannend an der Geschwindigkeit? Welchen Einfluss hat die Geschwindigkeit auf deinen Alltag? Welche Eigenschaften hat Geschwindigkeit?	Warum? Kannst du ein Beispiel dafür machen?	Das ist sehr spannend, was du sonst noch? Kannst du mir das noch genauer erklären. Was hast du über die Geschwindigkeit in der Schule gelernt?	-
Du siehst nun ein Scratchprogramm auf dem Bildschirm. Klicke auf die grüne Flagge und lies die Anleitung durch. Hast du eine Frage dazu? Bediene nun das Programm und denke dabei laut. Kannst du bitte alles sagen, was dir durch den Kopf geht. Klicke auf die Leertaste, um das Programm zu starten.	Wie funktioniert das Programm? Was ist das? (auf Variable Geschwindigkeit zeigen) Was macht das? Was hat die Geschwindigkeit mit diesem Programm zu tun?	Da hast du etwas gezögert/gelacht – woran hast du gedacht? Wieso hast du das gedrückt/geklickt?	Was siehst du? Was fällt dir auf? Was machst du gerade? Das ging sehr schnell. Was ist dir beim Bedienen des Programms durch den Kopf gegangen?	Programm Geschwindigkeit
Du siehst nun ein Scratchspiel auf dem Bildschirm. Klicke auf die grüne Flagge und lies die Anleitung durch. Hast du eine Frage dazu? Bediene nun das Programm und denke dabei laut. Kannst du bitte alles sagen, was dir durch den Kopf geht. Klicke auf die Leertaste, um das Programm zu starten.	Wie funktioniert das Programm? Was ist das? (auf Variable Geschwindigkeit zeigen) Was macht das? Was ist das? (auf Variable Punktestand zeigen) Was macht das? Was hat die Geschwindigkeit mit diesem Programm zu tun?	Da hast du etwas gezögert/gelacht – woran hast du gedacht? Wieso hast du das gedrückt/geklickt?	Was siehst du? Was fällt dir auf? Was machst du gerade? Das ging sehr schnell. Was ist dir beim Bedienen des Programms durch den Kopf gegangen?	Programm Geschwindigkeit Spiel

Konzept Variable

Leitfrage (Erzählaufforderung)	Subfragen / Nachfragen (auf Antworten reagieren)	Frage zur Begründung	Aufrechterhaltungs- und Steuerungsfragen	Material
Du hast mir bereits viel Spannendes erzählt. In unserem Gespräch kamen verschiedene Variablen vor. Hast du eine Idee, was eine Variable sein könnte?	<p>Es macht nichts, wenn du nicht weisst, was das ist. Du kannst mir deine Vermutung erzählen. Was könnte das sein?</p> <p>Die Variablen kamen auch in den Programmen und Spielen vor, die wir angeschaut haben.</p> <p>Was haben Zeit, die Temperatur und die Geschwindigkeit in den Programmen gemeinsam?</p> <p>Welches Spiel ist dir gut in Erinnerung geblieben? In diesem Spiel waren xxx und xxx die Variablen. Was haben sie in diesem Programm gemacht?</p>	<p>Warum?</p> <p>Woran erkennst du das?</p>		

Vielen Dank für die Zeit, welche du dir für das Interview genommen hast. Deine Ideen waren sehr spannend und helfen mir, besser zu verstehen, wie Kinder in deinem Alter denken. Herzlichen Dank!

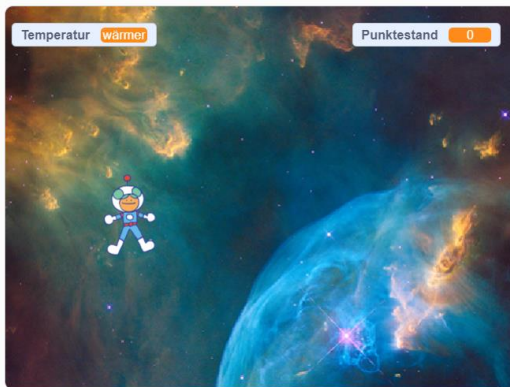
10.3.3 Beschreibung der Scratchprogramme

Temperatur Alltag

In diesem Programm muss die Katze mit der Maus durch verschiedene Lebensräume gesteuert werden. Oben links ist die Temperatur zu sehen, die sich je nach Ort ändert. Rechts sind verschiedene Kleidungsstücke zu sehen, die der Katze durch Anklicken angezogen werden können. Die Katze gibt über Sprechblasen Rückmeldung, wenn die Kleidung nicht passt.

Link zu Programm: <https://scratch.mit.edu/projects/1003463658>

Temperatur Spiel



Dieses Spiel ist an ein analoges Versteckspiel angelehnt. Eine Astronautin fliegt mit der Maus gesteuert durch den Weltraum und muss einen versteckten Kristall suchen. Anhand der Temperaturanzeige kann abgelesen werden, wie nah der Kristall ist (heiss am nächsten und eiskalt am weitesten entfernt). Sobald der Kristall gefunden wurde, erscheint er auf dem Bildschirm, ein Ton

ertönt und der Punktestand wird um eins erhöht. Das Spiel endet nach 90 Sekunden und zeigt die erreichte Punktzahl in einer Sprechblase an.

Link zu Programm: <https://scratch.mit.edu/projects/1016650724>

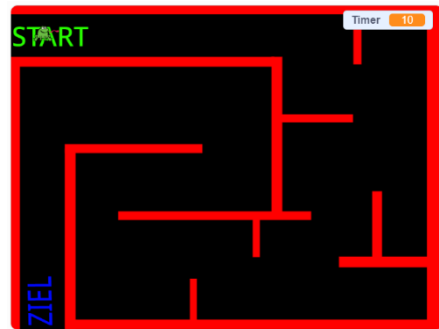
Zeit Alltag

In diesem Programm wird der Morgen der Katze geplant. Dazu werden verschiedene Fragen gestellt. Als Antwort müssen die Lernenden eine Uhrzeit eingeben, die dann über die Variable visualisiert und in der Sprechblase der Katze wiedergegeben wird.

Link zu Programm: <https://scratch.mit.edu/projects/1019586763>

Zeit Spiel

Bei diesem Spiel muss ein Frosch mit der Maus durch ein Labyrinth gesteuert werden. Sobald die rote Wand berührt wird, ist das Spiel beendet. Oben rechts wird die Variable Timer visualisiert. Am Ende zeigt die Sprechblase die benötigte Zeit an. Für dieses Spiel wurde eine Vorlage der PH Luzern verwendet. Genauer gesagt hat Linda Greter das ursprüngliche Projekt erstellt. Der Code wurde selbst erstellt und der Hintergrund leicht verändert.



Link zu Programm: <https://scratch.mit.edu/projects/994259043>

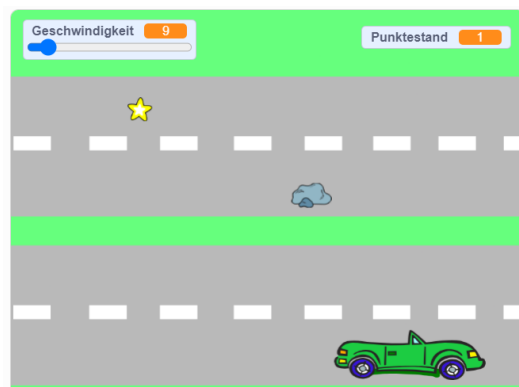
Geschwindigkeit Alltag

Bei diesem Programm fährt ein Auto auf einer Strasse von links nach rechts. Sobald es am rechten Strassenrand nicht mehr zu sehen ist, geht es zurück an den linken Strassenrand. Bei diesem Programm wurde eine Variable gewählt, die mit einem Schieberegler verändert werden kann. Je höher die Zahl, desto schneller fährt das Auto.

Link zu Programm: <https://scratch.mit.edu/projects/1013171307>

Geschwindigkeit Spiel

Dieses Spiel baut auf dem vorhergehenden Projekt auf. Die Steuerung der Geschwindigkeit bleibt gleich. Eine weitere Strasse wurde hinzugefügt. Das Auto kann nun mit den Pfeiltasten nach oben und unten die Spur wechseln. In zufälligen Abständen erscheinen Sterne, die Pluspunkte geben, und Steine, die Minuspunkte geben. Das Spiel endet nach 45 Sekunden und zeigt die erreichte Punktzahl in einer Sprechblase an.



Link zu Programm: <https://scratch.mit.edu/projects/1017401083>

10.3.4 Stichprobe U3

Nr.	Klasse	Alter	Ge- schlecht	Aufnahme	Ablauf	Zeit	Bemerkungen	Programmiererfahrung
01	JZ	12	m	KI_01	TGZ	37'	Variablen im Text erkannt	Früher zuhause mit Mutter, jetzt nicht mehr
02	JZ	13	m	KI_02	TZG	39'	LRS – Texte vorgelesen	Nur Schule (keine Erinnerung)
03	JZ	12	w	KI_03	ZTG	30'	Spiel mit Touchscreen bedient	Nur Schule (keine Erinnerung)
04	JZ	12	w	KI_04	GZT	32'	Bezeichnet Variable als Programmierer vom Auto	Nur Schule
05	JZ	12	m	KI_05	GTZ	33'	Sagte selbst, er war am Anfang sehr nervös.	Ein zusätzliches Projekt gemacht
06	LM	11	m	LM06	TGZ	41'	Viel Programmiererfahrung, Variable 2 x erwähnt. Kennt auch Verbindungs- block	Programmiert eigene Spiele
07	LM	12	m	LM_07	ZGT	33'	Variable bereits während erwähnt. Bei Temperatur konnte ich nicht alles Fra- gen. Kennt so vieles von Schwester und von Rodrigo (Kollege)	Mit anderen Kindern aus der Klasse ei- gene Spiele programmiert
08	LM	12	w	LM_08	TZG	41'	Zeitspiel vergessen und als letztes noch nachgeholt.	Spielt in der Freizeit nur
09	LM	12	m	LM_09	GZT	53'	Sehr gesprächig. Teils Fragen aus Zeitgründen weggelassen. Wort Operator für Block erwähnt.	Spiel mehr und sonst programmieren ohne Ziel
10	LM	12	M	LM_10 + LM_10b	GTZ	37'	Beschreibt Variable, Erklärt Variable bei Temperaturspiel ganz genau. Genau hinhören, weil sprachlich nicht alles verständlich. Testet bei Uhrzeit aus, fal- sche Uhrzeit und dann Baum. Beschreibt auch Pinkte Blöcke.	Aktuelles Schulprojekt, Freizeit viel (richtige Spiele)

Anmerkung. Rot= nur schulische Programmiererfahrung, Gelb= wenig zusätzliche Programmiererfahrung, Grün=viel zusätzliche Programmiererfahrung

10.3.5 Kodierleitfaden U3

Hauptkategorie / Unterkategorie	Definition	Ankerbeispiel
Programmiererfahrung Diese Kategorie bezieht sich auf die spezifische Frage aus dem Leitfaden.		
Schule	Hinweise zu Programmiererfahrungen, welche im Kontext der Schule gesammelt wurden.	Das erste Mal, wo wir Scratch hatten, konnte ich es nicht so gut. Wir hatten vorne Plakate, die man nachbauen, also nach programmieren mussten. Dann musste man als Endprodukt unsere Namen in ein Scratchprojekt einfügen und am Schluss leuchtete halt der Name oder ging in verschiedene Positionen. (Interview 6, Pos. 12)
Ausserhalb der Schule	Hinweise zu Programmiererfahrungen, welche in der Freizeit gesammelt wurden.	Ja, Ich habe mit X zusammen auch so ein Spiel programmiert. Ähm, so etwas ähnliches wie Cookie Clicker. Also, dass man einfach immer auf etwas drauf klicken muss und dann Upgrades eigentlich kann kaufen. (Interview 7, Pos. 14)
Zeit Hier wird die Beschreibung zur Zeit / die Äusserungen zur Zeit codiert. Es werden nur die Äusserungen vor dem Spielen der Scratchprogramme codiert.		
Uhrzeit	Das Kind beschreibt, dass man die Zeit auf einer Uhr erkennen kann oder dass es Uhrzeiten gibt.	Und ja, man sieht die Zeit auch überall auf dem Handy kann man die nachschauen, auf einer Uhr, auf einem PC. (JZ-05, Pos. 116)
Messung der Zeit	Das Kind beschreibt, wie Zeit gemessen werden kann. Das Kind beschreibt verschiedene Zeiteinheiten.	Die Zeit stoppen kann man auch. (JZ-04, Pos. 70) Also es hat halt Sekunden also Millisekunden Sekunden, Minuten, Stunden und und noch mehr dann ja eine Minute sind 60 Sekunden. Und 60 Minuten sind eine Stunde und einen Tag sind 24 Stunden. (LM-06, Pos. 118)
Wahrnehmung der Zeit	Kinder beschreiben, wie sie die Zeit wahrnehmen und empfinden. Themen wie Zeitdruck können auch vorkommen.	Manchmal stresst sie mich auch einbisschen. Weil ich halt nur wenig Zeit habe. Manchmal. Und auch noch Zeit für anderes haben will. (LM-07, Pos. 24)
Zeitkontinuität	Das Kind beschreibt, dass die Zeit unaufhaltsam weiter läuft. Es beschreibt, dass die Zeit schon immer da ist.	Ja, die kannst du eigentlich nicht anhalten. Sie läuft eigentlich immer weiter. (LM-07, Pos. 26)
Einfluss Alltag	Das Kind beschreibt, dass die Zeit einen Einfluss auf den Alltag hat.	Ohne die Zeit wüsste ich nicht, wann ich in die Schule muss. (...) Wann ich schlafen gehen muss und wann wo ich hingehen muss. (JZ-02, Pos. 101)

Temperatur			
Hier wird die Beschreibung zur Temperatur / die Äusserungen zur Temperatur codiert. Es werden nur die Äusserungen vor dem Spielen der Scratchprogramme codiert.			
Temperaturskalen		Das Kind erwähnt, dass Temperatur in Grad Celsius oder in Fahrenheit gemessen werden kann.	die Temperatur gibt einfach das Grad an, also wie warm das ist ist oder wie kalt. (LM-08, Pos. 16)
Messen der Temperatur		Das Kind beschreibt, dass Temperatur gemessen werden kann.	Man kann es in einem Thermometer messen. (Interview 1, Pos. 24)
Zustand Temperatur		Das Kind beschreibt den Zustand von Temperatur. Es beschreibt, dass Temperatur warm oder kalt sein kann.	Also Temperatur so, wie warm es ist oder wie kalt. Oder wie warm etwas ist. (JZ-02, Pos. 24)
Wetter		Das Kind beschreibt ein Wetterphänomen wie Wind oder Regen.	Zum Beispiel im Winter ist die Temperatur meistens so zwischen zehn und minus irgendwas, glaube ich. Und dann entsteht auch Schnee, zum Beispiel von der Temperatur. Oder das Wasser wird zu Eis (JZ-05, Pos. 74)
Veränderlichkeit		Das Kind beschreibt, dass sich die Temperatur verändern kann.	Das sie sich halt immer verändert. (LM-07, Pos. 136)
Einfluss von Temperatur	Einfluss Kleidung	Das Kind beschreibt, dass die Temperatur, die Kleidung beeinflusst, welche es anzieht.	Aber einfach ziehe ich mich noch besser um, dass mir nicht kalt ist. Oder wenn einfach heiss ist, gehe ich in kurzen Hosen und kurzen T-Shirt Fussball spielen. (Interview 6, Pos. 26)
	Einfluss Aktivität	Das Kind beschreibt, dass die Temperatur die Aktivitäten beeinflusst, welche es macht.	Schwimmen, wenn es warm ist. (.....) Wenn es kalt ist, halt so Ski fahren. Wenn es heiss ist auch so, auf dem Pumptrack oder auf dem Fussballplatz. (Interview 1, Pos. 32-33)
	Einfluss Erde/Welt-raum	Das Kind beschreibt den Einfluss der Temperatur auf die Erde. Das Kind beschreibt unterschiedliche Temperaturgebiete auf der Erde oder im Weltraum	Und auch bestimmen, wie es halt aussieht in der Gegend wo es kalt oder warm ist. (JZ-01, Pos. 34)
	Einfluss auf den Körper	Das Kind beschreibt, welchen Einfluss die Temperatur auf den Körper des Menschen haben kann.	Und ja, es ist sicher, man kann erfrieren an der Temperatur, wenn es zu tief ist im Minus. Und man kann auch verbrennen, wenn die Temperatur zu warm ist. Also im Plus zu viel. (JZ-05, Pos. 72)
Geschwindigkeit			
Hier wird die Beschreibung der Geschwindigkeit / die Äusserungen zur Temperatur codiert. Es werden nur die Äusserungen vor dem Spielen der Scratchprogramme codiert.			

Masseinheiten von Geschwindigkeit		Das Kind beschreibt, dass Geschwindigkeit in km/h oder m/h angegeben wird.	Und Geschwindigkeit misst man eigentlich meistens in km/h glaube ich. Also Kilometer die Stunde und. Ja. (Interview 7, Pos. 72)
Zustand von Geschwindigkeit		Das Kind beschreibt, dass etwas schnell oder langsam sein kann. Das Kind nutzt andere Wort für Geschwindigkeit wie Tempo oder Schnelligkeit.	Es ist spannend, weil man sehr schnell sein kann oder sehr langsam. (Interview 1, Pos. 91)
Geschwindigkeit bei unterschiedlichen Fortbewegungsmitteln	Autos	Das Kind spricht über Autos in Verbindung mit Geschwindigkeit.	Es ist die Tempoanzeigen im Auto. Wie schnell man halt fahren... fährt... und es ist halt auch bei der Autobahn oder... bei... Hauptstrassen gekennzeichnet, wie schnell man fahren darf. (Interview 2, Pos. 165)
	Laufen / Rennen	Das Kind spricht über Laufen oder Rennen in Verbindung mit Geschwindigkeit	Und auch beim Laufen dort hat man auch eine gewisse Geschwindigkeit und beim Fallen oder beim Fliegen auch. (Interview 8, Pos. 84)
	Computer	Das Kind spricht über Computer oder Handy in Verbindung mit Geschwindigkeit. Zu dieser Kategorie zählt auch, wenn über Computerprogramme und darin vorkommende Geschwindigkeiten gesprochen wird.	Es gibt auch Geschwindigkeit beim Computer oder beim Handy mit dem Internet. Zum Beispiel, wenn man ein Abo haben will, dass man überall WLAN hat. Dann gibt es manchmal auch Geschwindigkeit. Wenn man nicht ein so teures Abo kauft, geht es halt auch länger, bis die Geschwindigkeit, ähm bis die Seite geladen ist. (Interview 8, Pos. 84)
	Verschiedene	Das Kind benennt andere Fortbewegungsmittel in Verbindung mit Geschwindigkeit	Also beim Skifahren ist es halt so... die... Skifahrer tragen, glaube ich, ein GPS am Schuh. Und so wird es dann gemessen. (Interview 2, Pos. 167)
Definition Variable 1 Diese Kategorie bezieht sich auf die Beschreibung von Variablen nach dem Spielen der Programme mit Rücksicht auf die unterschiedlichen Hinweisniveaus.			
Erzählaufforderung + Vermutung		In diese Kategorie gehört alles, was das Kind nach der ersten Erzählaufforderung sowie einer möglichen Nachfrage erzählt.	-
Hinweis 1		Zu dieser Kategorie gehört alles, was das Kind nach dem ersten Hinweis erzählt.	-
Hinweis 2		Zu dieser Kategorie gehört alles, was das Kind nach dem zweiten Hinweis erzählt.	-
Hinweis 3		Zu dieser Kategorie gehört alles, was das Kind nach dem dritten Hinweis erzählt.	-

Definition Variable 2		
Diese Kategorie bezieht sich auf die Beschreibung von Variablen nach dem Spielen der Programme. Hier werden nur inhaltliche Kategorien berücksichtigt.		
Variablen als «verschiedene Sachen»	Das Kind beschreibt Variablen mit den Worten unterschiedliche oder verschiedene Sachen. Das Kind meint damit, dass Variablen verschiedene Sachen sein können.	Unterschiedliche Sachen, also wie Varianten. (Interview 1, Pos. 183)
Variablen sind Zahlen	Das Kind beschreibt, dass Variablen Zahlen sind oder Zahlen beinhalten.	Dass sie Zahlen haben, also wo man etwas eingeben muss, oder. Also ja sie haben Zahlen. (Interview 5, Pos. 158)
Variablen sind veränderbar	Das Kind beschreibt, dass Variablen veränderbar sind oder dass der Wert der Variable veränderbar ist.	Es waren Variablen, die sicher verändern, wenn etwas passiert und nachher zeigen sie, wie viele Mal, dass das passiert ist. Zum Beispiel Timer ist jede Sekunde passiert, verändert sich. Bei den Temperaturen ist jeder Raum verändert sich die Temperatur. Beim Warm heiss wärmer heiss. (Interview 10, Pos. 190)
Temperatur, Zeit und Geschwindigkeit sind Variablen	Das Kind beschreibt, dass Temperatur, Zeit oder Geschwindigkeit eine Variable war.	Vielleicht hier mit diesen hier [zeigt auf Variable im Programm]. Es hat ja bei jedem Spiel so etwas mit der Geschwindigkeit oder Temperatur. So etwas vielleicht mit diesen zu tun. (Interview 3, Pos. 182)
Variablen können Sachen speichern	Das Kind beschreibt, dass in Variablen etwas gespeichert werden kann.	Mit denen kann man Sachen wie abspeichern. (Interview 10, Pos. 221)
Variablen beschreiben etwas	Das Kind beschreibt, dass Variablen etwas beschreiben oder auch etwas anzeigen.	Sie beschreiben etwas Also man kann mit ihnen etwas beschreiben. Also ich würde sagen, das Auto fährt 30 km/h und ist 20 Grad warm. (Interview 2, Pos. 239)
Variablen können etwas auslösen	Das Kind beschreibt, dass Variablen einen Einfluss haben können.	Das waren alles Variablen. (..) Also in Variablen angegeben hier. Und die hatten alle etwas mit dem Spiel zu tun, also hatten eigentlich einen Einfluss. Also ob es jetzt warm ist oder kalt oder wie schnell das Auto fahren muss oder. Ähm. Was war das Letzte noch. Die Zeit. Wie lange du hattest. (Interview 7, Pos. 192)
Variablen als vielseitiges Programmierkonzept	Das Kind beschreibt, was man mit Variablen programmieren kann	Äh Variablen kann man eben. Das benutzt man eher in schwierigeren Spiele mit mehr Variablen, weil dort kann man wie alles. Wie viele Sterne hast du wo. Bei zum Beispiel ich habe Tower Defense mal gemacht und nicht fertig ist. Dort kann man bei jedem Clon schauen, wo er ist und wer am weitesten ist und so. Ja und ohne Variablen kann man das alles nicht machen. (Interview 10, Pos. 214)
Variablen werden benannt	Das Kind beschreibt, dass Variablen benannt werden	Die sind auch immer benannt eigentlich und dann weiss man immer genau, was das ist. (Interview 7, Pos. 186)

Eine Variable ist eine Box	Das Kind beschreibt, dass Variablen wie eine Box sind	Eine Variable ist dann wieso ein Kasten oder eine Box, in dem etwas steht, was man also was, was das ist. (Interview 8, Pos. 142)
Variablen sind nicht bestimmbar	Das Kind beschreibt, dass Variablen etwas unbestimmbares sind.	Etwas, das man nicht wirklich selbst bestimmen kann. (Interview 1, Pos. 197)

10.2.8 – 10.2.10 Interviewtranskript 1-3

Für Veröffentlichung entfernt.