

# Die Kompetenz «Algorithmen» aus dem Lehrplan 21

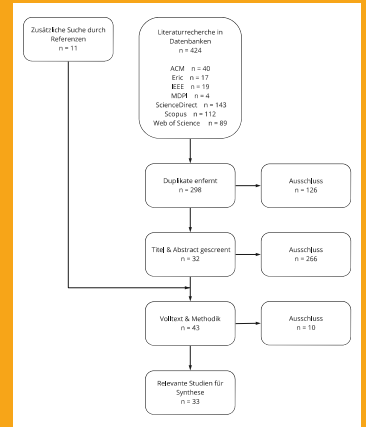
Ein systematisches Literaturreview zur Kompetenzüberprüfung mit der Programmiersprache Scratch

## Abstract

Die Kompetenz «Algorithmen» aus dem Kompetenzbereich Informatik des Lehrplans 21 beschreibt Problemlösestrategien und beinhaltet zentrale Konzepte zur Algorithmisierung und Programmierung. Bezüglich der kompetenzorientierten Überprüfung dieser Inhalte existiert bisher wenig empirisch fundierte Expertise. Aus diesem Grund interessiert die Frage, wie die Inhalte der Kompetenz «Algorithmen» im Schulalltag überprüft werden können. Dazu wurde in einem ersten Schritt ein theoretischer Rahmen erstellt, welcher die Inhalte ausdifferenziert und unter dem Aspekt der Kompetenzorientierung verordnet. Auf dieser Basis wurde in einem nächsten Schritt ein systematisches Literaturreview durchgeführt, um eine Übersicht der Überprüfungsmöglichkeiten der Inhalte sowie der Operationalisierung zu gewinnen.

Daraus resultiert, dass die Operationalisierung der Inhalte mehrheitlich auf Basis von Scratchblöcken vorgenommen wird. Am häufigsten werden Scratchprojekte summativ und hinsichtlich der Komplexität überprüft. Eine grosse Forschungslücke besteht zu formativen Überprüfungsmöglichkeiten. Insbesondere gilt es Feedbackmethoden zu untersuchen, da diese für formative Beurteilungen zentral sind.

Fragestellungen	Methode
<ul style="list-style-type: none"> <li>F1: Welche Computational Thinking (CT) -Komponenten liegen der Kompetenz «Algorithmen» aus dem Lehrplan 21 in Bezug zur Programmierung zugrunde?</li> <li>F2a: Wie werden die einzelnen CT-Komponenten bei Schülerinnen und Schülern in der obligatorischen Schulzeit mit der Programmiersprache Scratch operationalisiert?</li> <li>F2b: Wie werden die CT-Kategorien Algorithmisches Denken und Evaluieren formativ mit der Programmiersprache Scratch überprüft?</li> </ul>	<ul style="list-style-type: none"> <li>Systematisches Literaturreview</li> <li>Durchführung: 11.10.2021</li> <li>Fokus der Studien auf Programmierfähigkeiten und mit Scratch überprüft</li> <li>Zielgruppe 6– 16 Jahre</li> <li>Suchfilter: "computational thinking" AND scratch AND (assess* OR test*)</li> </ul>



Flussdiagramm systematisches Literaturreview

Ergebnisse	Schlussfolgerungen / Diskussion																																																																																				
<p><b>F1: Theoretischer CT-Rahmen der Kompetenz «Algorithmen»</b></p> <table border="1"> <thead> <tr> <th>Kompetenzkassen</th> <th>Deklaratives Wissen</th> <th colspan="2">Prozedurales Wissen</th> </tr> <tr> <th>CT Kategorien</th> <th>Konzeptionelles Wissen</th> <th>Algorithmisches Denken</th> <th>Evaluation</th> </tr> <tr> <th>CT Komponenten</th> <th>Abläufe, Schichten, bedingte Anweisungen, Parameter, Variablen, Unterprogramme</th> <th>Abstraktion, Dekomposition</th> <th>Testen, Debugging, Evaluieren</th> </tr> </thead> <tbody> <tr> <td>MI.2.2</td> <td></td> <td>Lösungswege formulieren; Lösungen in Programmen umsetzen</td> <td></td> </tr> <tr> <td>MI.2.2.d</td> <td>Abläufe; bedingte Anweisungen; Schichten; Parameter</td> <td>Lösungswege formulieren</td> <td></td> </tr> <tr> <td>MI.2.2.e</td> <td>Anweisungen; Programm ist Abfolge von Anweisungen</td> <td></td> <td></td> </tr> <tr> <td>MI.2.2.f</td> <td>Programm; Abfolge</td> <td></td> <td>Programme testen</td> </tr> <tr> <td>MI.2.2.g</td> <td>Bedingte Anweisung; Parameter; Schichten</td> <td>Lösungswege formulieren</td> <td></td> </tr> <tr> <td>MI.2.2.h</td> <td>Algorithmen; Variablen; Unterprogramme</td> <td>Algorithmen formulieren</td> <td></td> </tr> </tbody> </table> <p><small>Theoretischer Rahmen mit den Inhalten aus den Kompetenzstufen der Kompetenz «Algorithmen» aus dem LP 21.</small></p>	Kompetenzkassen	Deklaratives Wissen	Prozedurales Wissen		CT Kategorien	Konzeptionelles Wissen	Algorithmisches Denken	Evaluation	CT Komponenten	Abläufe, Schichten, bedingte Anweisungen, Parameter, Variablen, Unterprogramme	Abstraktion, Dekomposition	Testen, Debugging, Evaluieren	MI.2.2		Lösungswege formulieren; Lösungen in Programmen umsetzen		MI.2.2.d	Abläufe; bedingte Anweisungen; Schichten; Parameter	Lösungswege formulieren		MI.2.2.e	Anweisungen; Programm ist Abfolge von Anweisungen			MI.2.2.f	Programm; Abfolge		Programme testen	MI.2.2.g	Bedingte Anweisung; Parameter; Schichten	Lösungswege formulieren		MI.2.2.h	Algorithmen; Variablen; Unterprogramme	Algorithmen formulieren		<p><b>F2a: Scratchblöcke zur Messung von Komplexität in Programmen</b></p> <table border="1"> <thead> <tr> <th>CT-Komponenten in Scratch</th> <th colspan="5">Komplexität</th> </tr> </thead> <tbody> <tr> <td>Ablauf</td> <td>Blöcke verwenden</td> <td>Blöcke mit spezifischer Ausgabe</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Schichten</td> <td>wiederhole, wiederhole fortlaufend</td> <td>wiederhole bis</td> <td>Wiederholung mit logischen Operatoren</td> <td>verschachtelte Wiederholungen</td> <td>verschachtelte Wiederholungen mit logischen Operatoren</td> </tr> <tr> <td>Bedingte Anweisungen</td> <td>falls, dann</td> <td>falls, dann sonst</td> <td>logische Operatoren</td> <td>verschachtelte logische Operatoren</td> <td></td> </tr> <tr> <td>Parameter</td> <td>Figureigenschaften modifizieren</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>Variablen</td> <td>verbinde-Block</td> <td>eigene Variable erstellen</td> <td>Listen verwenden</td> <td>Listen als Stapel verwenden</td> <td></td> </tr> <tr> <td>Abstraktion Unterprogramme</td> <td>eigener Block erstellen</td> <td>eigener Block mit Parameter</td> <td>Klöne erstellen</td> <td>Rekursion</td> <td></td> </tr> <tr> <td>Dekomposition</td> <td>mehrere Schritte als Teilsystem</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p><small>Differenzierung der CT-Komponenten mit Scratchblöcken</small></p> <p><b>Darstellung der Kompetenz «Algorithmen» in Schnittmengen</b></p> <p><small>Abbildung Kompetenz «Algorithmen»</small></p>	CT-Komponenten in Scratch	Komplexität					Ablauf	Blöcke verwenden	Blöcke mit spezifischer Ausgabe				Schichten	wiederhole, wiederhole fortlaufend	wiederhole bis	Wiederholung mit logischen Operatoren	verschachtelte Wiederholungen	verschachtelte Wiederholungen mit logischen Operatoren	Bedingte Anweisungen	falls, dann	falls, dann sonst	logische Operatoren	verschachtelte logische Operatoren		Parameter	Figureigenschaften modifizieren					Variablen	verbinde-Block	eigene Variable erstellen	Listen verwenden	Listen als Stapel verwenden		Abstraktion Unterprogramme	eigener Block erstellen	eigener Block mit Parameter	Klöne erstellen	Rekursion		Dekomposition	mehrere Schritte als Teilsystem				
Kompetenzkassen	Deklaratives Wissen	Prozedurales Wissen																																																																																			
CT Kategorien	Konzeptionelles Wissen	Algorithmisches Denken	Evaluation																																																																																		
CT Komponenten	Abläufe, Schichten, bedingte Anweisungen, Parameter, Variablen, Unterprogramme	Abstraktion, Dekomposition	Testen, Debugging, Evaluieren																																																																																		
MI.2.2		Lösungswege formulieren; Lösungen in Programmen umsetzen																																																																																			
MI.2.2.d	Abläufe; bedingte Anweisungen; Schichten; Parameter	Lösungswege formulieren																																																																																			
MI.2.2.e	Anweisungen; Programm ist Abfolge von Anweisungen																																																																																				
MI.2.2.f	Programm; Abfolge		Programme testen																																																																																		
MI.2.2.g	Bedingte Anweisung; Parameter; Schichten	Lösungswege formulieren																																																																																			
MI.2.2.h	Algorithmen; Variablen; Unterprogramme	Algorithmen formulieren																																																																																			
CT-Komponenten in Scratch	Komplexität																																																																																				
Ablauf	Blöcke verwenden	Blöcke mit spezifischer Ausgabe																																																																																			
Schichten	wiederhole, wiederhole fortlaufend	wiederhole bis	Wiederholung mit logischen Operatoren	verschachtelte Wiederholungen	verschachtelte Wiederholungen mit logischen Operatoren																																																																																
Bedingte Anweisungen	falls, dann	falls, dann sonst	logische Operatoren	verschachtelte logische Operatoren																																																																																	
Parameter	Figureigenschaften modifizieren																																																																																				
Variablen	verbinde-Block	eigene Variable erstellen	Listen verwenden	Listen als Stapel verwenden																																																																																	
Abstraktion Unterprogramme	eigener Block erstellen	eigener Block mit Parameter	Klöne erstellen	Rekursion																																																																																	
Dekomposition	mehrere Schritte als Teilsystem																																																																																				

Auswertung systematisches Literaturreview:



- Kompetenzüberprüfung beinhaltet prozedurales Wissen
- Kaum Forschungen zu prozeduralem Wissen resp. formativer Überprüfung (F2b)
- Weitere Forschung insbesondere im Bereich Feedbackmethoden notwendig.

Verwendete Literatur: