

Klaus-Tycho FÖRSTER, Zürich

## **Scratch von Anfang an: Programmieren als begleitendes Werkzeug im mathematischen Unterricht der Sekundarstufe**

Die Programmierung ist heutzutage weitgehend aus dem mathematischen Unterricht verschwunden (vergl. etwa Oldenburg 2011), eine nicht nur nach Kortenkamp 2005 bedauerliche Entwicklung, denn „*Konzepte wie Schleifen, Prozeduren und insbesondere Variablen sind eigentlich unabdingbar*“. Gerade für die fundamentale Idee der Algorithmen ist die Programmierung hilfreich, insbesondere für die Überprüfung auf Korrektheit und formale Präzisierung. Die Wichtigkeit von Algorithmen ist nicht auf die Mathematik und Informatik beschränkt: Algorithmen sind „*fächerübergreifend und alltagsrelevant*“ (Schmidt-Thieme 2005). Zudem unterstützt die Programmierung beim Problemlösen, denn sie bietet eine gemeinsame formale Sprache. Somit kann man über Programme leichter reden: Über ihre Struktur, ihre Entwicklung und ihre Beziehungen zu anderen Problemen und Programmen (siehe u.a. Feurzeig und Papert 1969).

Analog zu einer Funktion führt ein Computerprogramm eine Transformation von einer Eingabe (Argument) zu einer Ausgabe (Wert) durch (vergl. etwa Puhmann 1998). Somit kann die Aufgabe, im Unterricht ein Programm zu schreiben, mit einer Beweisaufgabe verglichen werden. Gegebenes und Gesuchtes sind bekannt, die Transformation aber nicht, siehe Streckler 2009. Der Entwurf eines korrekten Programms entspricht einem direkten (konstruktiven) Beweis. Da Beweisaufgaben im Mathematikunterricht leider unterrepräsentiert sind (z.T. nur 1%, Neubrand 2002), bietet sich die Programmierung unmittelbar für das alltägliche Unterrichtsrepertoire an.

### **1. Programmierung mit Scratch im MU: Begleitend und unterstützend**

Eine Integration der Programmierung in den mathematischen Unterricht durch zusätzlichen Zeitaufwand ist kaum realisierbar: Die Reduktion bewährter mathematischer Inhalte oder gar anderer Fächer würde – zu Recht – auf starken Widerstand von allen Seiten stoßen. Daher bietet sich für den Schulalltag (abgesehen von Projekten) nur eine begleitende Integration an.

Einige zusätzliche Anwendungsbereiche der Programmierung wurden in den letzten Jahrzehnten auch von Spezialprogrammen (z.B. DGS, CAS, Tabellenkalkulation) übernommen: Dies ist sicherlich eine hilfreiche spezifische Anwendungsprogrammierung, aber der gewünschte vertiefte konstruktive Entwurf und seine Formalisierung treten hier notwendigerweise in den Hintergrund.

In J. Roth & J. Ames (Hrsg.), *Beiträge zum Mathematikunterricht 2014* (S. 373–376).  
Münster: WTM-Verlag

Programmierung ist nicht als Wert an sich zu verstehen und auch nicht als Konkurrent zu anderen computergestützten Anwendungen, wie etwa dynamischer Geometriesoftware. Vielmehr gilt es die Chancen der Programmierung für vielfältige Einsatzbereiche zu nutzen: Problemlösen, Beweisen, Fachsprache, Kritikfähigkeit, Modellierung und insbesondere für das grundlegende algorithmische Konzept der Modularisierung. Der begleitende Einsatz schließt sich an bewährte Konzepte der GDM von 1981 an, die einen Computereinsatz durch methodische Schwerpunktverlagerungen und Änderungen im Kleinen empfiehlt, bzw. gemäß Kortenkamp 2005: „*durch die Anwendung einer konkreten Programmiersprache, wann und wo immer es im Unterricht hilfreich erscheint*“. Nach Kortenkamp 2005 erscheint Programmierung auch für die Grundschule „*durchaus gerechtfertigt*“.

Im Gegensatz zu vielen anderen Programmiersprachen benötigt Scratch ([scratch.mit.edu](http://scratch.mit.edu)) durch seinen visuellen Ansatz kaum Einarbeitungszeit und vermeidet durch Programmierung analog zu einem Baukastenprinzip Syntaxfehler. Die Programme selbst können leicht verglichen und analysiert werden. Lehramtsstudierende aller Fächer können mit Scratch schon in den ersten Semestern mit Schülern sogar der ersten Klasse erfolgreich programmieren (Förster 2011). Ebenso wurde Scratch mit seinen Erweiterungen Snap/BYOB auch schon an einzelnen deutschen Schulen bis zum Zentralabitur Informatik und zur Programmierung im Studium eingesetzt.

## **2. Begleitender Einsatz von Programmierung mit Scratch in der Praxis**

Eine begleitende Integration der Programmierung in den MU mit Scratch darf keine Unterrichtsinhalte vorwegnehmen. Daher sollten vor Klassenstufe 7 Variablen nicht eingesetzt werden. Von da an könnte eine Betrachtung dieser aus Programmiersicht neue Möglichkeiten des Variablenverständnisses eröffnen (siehe Serafini 2011). Dies ist vor allem dadurch von Interesse, da u.a. „*der richtige Umgang mit Variablen ... die wesentliche Voraussetzung für das Verständnis der weiterführenden Mathematik [darstellt]*“ (Reimann 2011). Für bewährte Konzepte lässt sich auf die Turtlegrafik mit Logo zurückgreifen, welche nach Hromkovič 2012 eine „*gegenseitige Befruchtung mit dem Geometrieunterricht*“ leicht erreicht und „*die Entwicklung algorithmischen Denkens*“ prägt. Bei einem Unterrichtsversuch über die Konstruktion von Vielecken und Parketten in der Klassenstufe 6 ergab sich, dass die Einführung in die Programmiersprache Scratch intuitiv und unproblematisch war, die Konzepte der Turtlegrafik sich direkt umsetzen lassen, Vielecke den Bereich Konzept und Einsatz von Algorithmen üben und verdeutlichen, sowie Parkette die Bedeutung von Exaktheit aufzeigen und modularen Entwurf benötigen (Förster 2013).

Für eine Fortführung des Einsatzes der Programmierung in Klassenstufe 7 mit derselben Klasse (12 Mädchen, 12 Jungen) bieten sich die Konstruktionsbeschreibungen von Dreiecken an, denn „*Konstruktionsbeschreibungen im Geometrieunterricht haben als Endform den Algorithmus, der dann in Computersprachen übersetzt werden kann*“ (Schmidt-Thieme 2009). Ebenso gestatten Konstruktionsbeschreibungen mit Scratch einen intuitiven Variableneinsatz, wie dies die untere Abbildung illustriert.



Schülerprogramm für die Konstruktion Seite-Winkel-Seite, links für eine spezielle Instanz, auf der rechten Seite dann allgemein. Propädeutisch ist dies sogar eine Funktion von drei Veränderlichen.

Nach Weigand und Weth 2002 ermöglicht die Realisation durch den Computer „*drastische Verbesserungen beim Beschreiben von Konstruktionen*“ durch Umkehrung der Reihenfolge „*erst die Konstruktion, dann die Beschreibung*“. Ebenso führt Riemer 2011 aus, dass Schüler nur widerwillig Konstruktionsbeschreibungen durchführen und „*‘Welten‘ zwischen Schülerprodukten und fachsprachlich akzeptablen Lösungen liegen*“. Er regt hierzu als Verbesserung an, dass „*ein Perspektivwechsel, ein frühes ‚digitales Nachdenken über händisches Tun‘ faszinierende Möglichkeiten gedanklicher Vertiefung und bisher noch wenig ausgetretene Pfade zu einer höheren mathematischen Bewusstheit [bietet]*“.

Aus diesem Grund entschieden wir uns für den folgenden dreigliedrigen Ansatz für die Unterrichtseinheit: Zuerst die Programmierung von Dreieckskonstruktionen in Scratch (gleichseitiges Dreieck, Seite-Winkel-Seite, WSW, Dreiecksscharen), dann die Konstruktion in GeoGebra (u.a. mit SSW&SSS) und abschließend Konstruktion und Konstruktionsbeschreibung per Hand. Dabei bietet die Programmierung insbesondere den Vorteil, dass man nicht mehr eine spezielle Instanz des Problems konstruiert, sondern alle Variationen der Problemstellung gleichzeitig als Algorithmus erstellt. Es ergab sich, dass die Schüler mit Scratch nach kurzer Einführung selbständig die Programme erstellen und vor allem auch gemeinsam darüber reden konnten, sei es nun im Plenum oder in der Arbeitsphase. Die Programmierung bot einen anderen Blick auf die bekannten Variablen (z.B. Änderung zur Laufzeit bei Dreiecksscharen), wobei es den Schülern durch das bekannte Thema (Geometrie) leicht fiel, die Variablen sinnvoll zu verwenden. Zudem wurde ein propädeutischer Blick auf die Funktionsweise von DGS vermittelt. Im Vergleich zu einem Vorgehen ohne Programmie-

rung zeigte sich deutlich, dass die Konstruktionsbeschreibungen akkurater waren, mit insbesondere mit weniger überflüssigem textuellen Beiwerk.

Wir erproben gegenwärtig Scratch in verschiedenen Bereichen begleitend im Mathematikunterricht (für eine Liste weiterer Möglichkeiten siehe Förster 2013) und werden hierüber an anderer Stelle ausführlicher berichten. Für einen interessanten aktuellen Einsatz zum Sinus- & Kosinussatz sei auf Untersuchungen an der Universität Hildesheim von Spittel 2014 verwiesen.

## Literatur

- Feurzeig, W., Papert, S. (1969): Programming-languages as a conceptual framework for teaching mathematics. In: Programmed Learning Research: Paris: Dunod.
- Förster, K.-T. (2011): Neue Möglichkeiten durch die Programmiersprache Scratch: Algorithmen und Programmierung für alle Fächer. In: BzMU 2011, 263-266.
- Förster, K.-T. (2013): Die Programmiersprache Scratch im Mathematikunterricht der Sekundarstufe I. In: BzMU 2013, 316-319.
- Hromkovič, J. (2012): Einführung in die Programmierung mit LOGO: Lehrbuch für Unterricht und Selbststudium. Wiesbaden: Springer Vieweg.
- Kortenkamp, U. (2005): Strukturieren mit Algorithmen. In: Kortenkamp et. al. (Hrsg.): Informatische Ideen im Mathematikunterricht. Hildesheim: Franzbecker, 77-85.
- Neubrand, J. (2002): Eine Klassifikation mathematischer Aufgaben zur Analyse von Unterrichtssituationen. Hildesheim: Franzbecker.
- Oldenburg, R. (2011): Mathematische Algorithmen im Unterricht: Mathematik aktiv erleben durch Programmieren. Wiesbaden: Vieweg+Teubner Verlag.
- Puhlmann, H. (1998): Funktionales Programmieren: eine neue Verbindung von Informatikunterricht und Mathematik. Darmstadt, Techn. Univ., FB Mathematik.
- Reimann, K. (2011): Probleme des Mathematikunterrichtes beim Übergang von Arithmetik zur Algebra. In: BzMU 2011, 671-674.
- Riemer, W. (2011): Erziehen im Mathematikunterricht. In: Kaenders, R., Schmidt, R. (Hrsg.): Mit GeoGebra mehr Mathematik verstehen, 13-20. Wiesbaden: Vieweg+Teubner Verlag.
- Serafini, G. (2011): Teaching Programming at Primary Schools: Visions, Experiences, and Long-Term Research Prospects. ISSEP 2011, 143-154.
- Schmidt-Thieme, B. (2005): Algorithmen – fächerübergreifend und alltagsrelevant? In: Engel, Joachim u. a. (Hrsg.): Strukturieren - Modellieren - Kommunizieren. Leitbilder mathematischer und informatorischer Aktivitäten. Hildesheim, 177-188.
- Schmidt-Thieme, B. (2009): Erklären als fachspezifische Kompetenz in fächerübergreifender Perspektive. In: BzMU 2009, 239-242.
- Strecker, K. M. (2009): Informatik für Alle – Wie viel Programmierung braucht der Mensch? Dissertation, Georg-August-Universität Göttingen.
- Spittel, L. (2014): Sinus- und Kosinussatz im Realschulunterricht unter Berücksichtigung der Geschichte und medialer Möglichkeiten. Masterarbeit, U. Hildesheim.
- Weigand, H.-G., Weth, T. (2002): Computer im Mathematikunterricht. Neue Wege zu alten Zielen. Heidelberg Berlin: Spektrum Verlag.