Zeitschrift: Bildungsforschung und Bildungspraxis : schweizerische Zeitschrift für Erziehungswissenschaft = Éducation et recherche : revue suisse des sciences de l'éducation = Educazione e ricerca : rivista svizzera di scienze dell'educazione

Herausgeber: Schweizerische Gesellschaft für Bildungsforschung

Band: 9 (1987)

Heft: 1

Artikel: Aspekte von LOGO

Autor: Rohrbach, Christian

DOI: https://doi.org/10.5169/seals-786347

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 12.07.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Aspekte von LOGO

Christian Rohrbach

- *Was soll der Computer in der Schule?*
  Die drei Bereiche CUU, Anwendersoftware und Programmieren, die für die Informatikbemühungen in der Volksschule relevant sind, werden kurz umrissen.
- *Warum trotzdem Programmieren?*
  Programmieren zeigt einen Aspekt des Computers auf, den die Anwendersoftware nicht abdeckt; er wird zum "Denkzeug".
- *Weshalb aber gerade LOGO?*
  LOGO ist eine interaktive Sprache mit muttersprachlichen Befehlswörtern und bietet über die vorzüglichen Grafikmöglichkeiten einen leichten Einstieg.
- *Was charakterisiert LOGO als didaktische Programmiersprache?*
  Die Igelgeometrie kann durchaus einen neuen Zugang zur Geometrie bedeuten. Daneben bietet LOGO viele Vorstellungshilfen für korrektes Verständnis mathematischer Sachverhalte.
- *Was darf von LOGO erwartet werden?*
  LOGO ist nicht ***das*** *Allerheilsmittel gegen alle möglichen schulischen Probleme. Aber LOGO ist motivierend und eines der sinnvollsten und hilfreichsten Instrumente für die Informatik in der Volksschule.*

Was soll der Computer in der Schule?

Bei der Frage, was der Computer in der Volksschule zu suchen habe, sind oft drei Bereiche anzutreffen, über die diskutiert wird:

1. Beim **Computer-unterstützten Unterricht** (CUU, CAI = Computer Assisted Instruction) - im eigentlichen (engen) Sinne verstanden als Ausprägung des Programmierten Unterrichts, wie er vor rund 30 Jahren von Skinner, Crowder u.a. in den USA begründet wurde - ist der Computer Trägermedium für die Darbietung des Unterrichts-Stoffes, der Übungsaufgaben usw.

   Dem CUU wird im zukünftigen Schulalltag häufig nur marginale Präsenz prophezeit. Allenfalls seien reine, aber flexibel verwendbare Übungs-/Drill-Programme beschränkt tauglich und nützlich. Gründe dafür mögen unter anderem sein, dass

- einerseits ein komplexes Programm, das auf viele allenfalls auftretende Probleme des Benutzers mittels Fehleranalyse gezielt eingeht, sehr hohe Entwicklungskosten verursacht, aber letzlich doch nie alle Schwierigkeiten abdeckt, also genügend umfassend sein kann,

- anderseits der Computer den Lernenden allzu sehr einengt, gängelt - in viel grösserem Masse noch als es etwa der Programmierte Unterricht in Buchform tut; die Antwortkontrolle z.B. erfolgt durch die Maschine und nicht durch den Lernenden selber; "zurückblättern" oder Teile aus eigenem Ermessen überspringen, "Vorschau halten" sind bei Lernprogrammen auf dem Computer kaum möglich.

2. Als die Computer immer kleiner und billiger wurden und als Folge davon auch in der Volksschule Eingang fanden, stand ganz eindeutig die **Tätigkeit des Progrmmierens** - notwendig verbunden mit dem Erlernen einer Programmiersprache (zumeist BASIC) - im Vordergrund der Aktivitäten in der Volksschule, wie sie die Pioniere unter den Lehrern initiierten. Heute ist man sich immer mehr einig darüber, dass Programmieren eine Fertigkeit ist, die kein allgemeines Bildungsgut und -ziel sein kann und also nicht jeder Volksschulabgänger beherrschen muss: Im anschliessenden Berufsleben werden nur ganz wenige zum/r Programmierer/in ausgebildet, und in nur sehr wenigen Berufen sind Programmierkenntnisse nützlich, geschweige denn erforderlich.

   Was an allgemeinen übergeordneten Zielen wie "logisch strukturiertes Denken", "analytisches Vorgehen" usw. allenfalls beim Programmieren-Lernen erreicht werden kann, kommt geradesogut auch beim Einsatz von

3. **Anwendersoftware** zum Zuge. Dabei geht es also darum, dass sich der Schüler in der Situation eines Computerbenützers befindet, so wie es einige - längst nicht die meisten - im späteren Beruf erleben werden. Dabei wird Standardsoftware, also nicht speziell für schulischen Gebrauch entwickelte Programme, für die Lösung von Problem- und Aufgabenstellungen aus verschiedenen Schulfächern eingesetzt. Viele Projekte auf der Volksschuloberstufe gehen von diesem Ansatz aus.

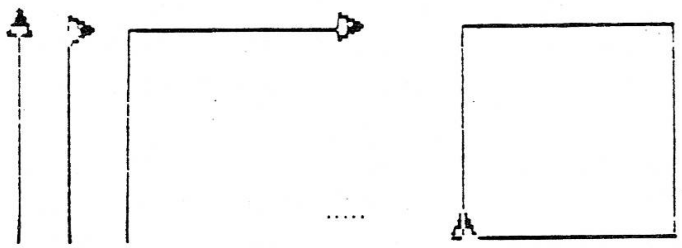
Warum trotzdem Programmieren?

LOGO ist eine Programmiersprache! Mit LOGO arbeiten heisst also letztlich "Programmieren lernen". Lohnt es sich da überhaupt darüber zu reden? Weshalb sollen jetzt Schüler gerade mit LOGO arbeiten statt z.B. mit einer anderen Programmiersprache oder noch viel besser mit schülergerechter Anwendersoftware?

Die Igelgrafik von LOGO

Bei der Igelgrafik wird durch LOGO-Befehlswörter ein meist dreieckiger Cursor (Zeichen auf dem Bildschirm) direkt, d.h. ohne Koordinatenangaben auf dem Bildschirm herumdirigiert. Z.B. erzeugt die Befehlsfolge

```
VORWÄRTS 70
RECHTS 90
VORWÄRTS 70
RECHTS 90
VORWÄRTS 70
RECHTS 90
VORWÄRTS 70
RECHTS 90
```



ein Quadrat.

Man kann

die acht Zeilen zusammenfassen:

```
WIEDERHOLE 4 [VORWÄRTS 70 RECHTS 90]
```

die Befehle abkürzen: `WH 4 [VW 70 RE 90]`

die Befehlsfolge unter einem Namen z.B. QUADRAT abspeichern, sodass später die Eingabe dieses Namens statt der Befehle genügt:

```
LERNE QUADRAT
WH 4 [VW 70 RE 90]
END
```

QUADRAT ist bereits ein (kleines) Programm, eine LOGO-Prozedur.

Unter der Prämisse, dass der Computer wie eigentlich jede Maschine dem Menschen dienen soll und nicht umgekehrt, ist der Trend weg von der Maschinensprache, weg auch von der bereits komfortablen höheren Programmiersprache hin zur benützerfreundlichen Anwendersoftware, in Zukunft vielleicht sogar hin zur Spracheingabe über's Mikrofon und nicht mehr über die Tastatur, nur folgerichtig. Allerdings bedeutet diese Quasi-Annäherung an den Menschen immer auch eine zunehmende Verfremdung, eine zunehmend grösser werdende Black-box um die Black-box, die die Hardware "Computer" selber schon ist.

Der Computer bleibt so ein sehr nützliches Werkzeug, mit dem rasch und effizient Texte bereinigt und schön dargestellt werden, mit dem schnell Statistiken ausgewertet, Tabellen erstellt und Diagramme gezeichnet werden, mit dem auf einfache Art ansprechende Grafiken und Titel gestaltet werden usw. Und dabei können die Schüler als Benützer immer lediglich die vorgesehenen Möglichkeiten nutzen; nie wird die Anwendersoftware so direkt zu kreativen Entdeckungen führen.

Der Computer unterstützt (rasch, sauber, exact, unermüdlich) das handwerkliche, technische Ausführen. Nie aber provoziert er aus den auftretenden, eben rein technischen Problemen sachimmanente Fragen; nie unterstützt er das Denken, wird also quasi zum eigentlichen "Denkzeug". Gerade diese Möglichkeiten des Computers, die über das rein Werkzeughafte hinausgehen, aber sind es, was viele Leute ängstigt, verunsichert. Die herkömmlichen Maschinen sind uns Krücken für die Fortbewegung, für die körperliche Arbeit, für unsere Sinne usw. Die Computer hingegen sind ebenfalls Krücken für etwas, sie "verstärken", ergänzen etwas, ja könnten in Konkurrenz treten zu etwas, das uns bisher als rein menschliches Vorrecht und Domäne galt: das Denken. Soll auch dieser Aspekt der "Computerfähigkeiten" bei den Informatikaktivitäten in der Volksschule berücksichtigt werden, muss der Schüler den Computer auch in dieser Hinsicht "erfahren" können. Dazu muss er der Maschine sagen können, was sie zu tun hat, wie sie sich "verhalten" soll: wenige einfache Kenntnisse in einer Programmiersprache sind dafür nötig.

Mit Anwendersoftware löst der Computerbenützer "Geschehen" auf dem Bildschirm aus, beim Programmieren bestimmt er es!

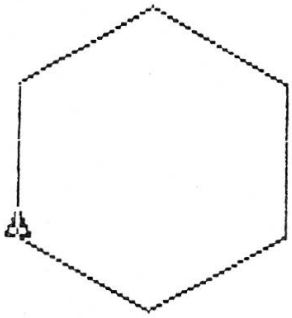
Weshalb aber gerade LOGO?

Von Volksschülern weitreichende Kenntnisse und Fertigkeiten in einer Programmiersprache zu verlangen, wäre sicher verfehlt. Programmier-Virtuosen sind nicht gefragt und Programmieren soll nicht zum Selbstzweck degenerieren. Darum lautet die Frage nicht "mit welcher Programmiersprache kann in der Schule am professionellsten gearbeitet werden?" Übrigens lässt sich wohl mit jeder höheren Programmiersprache jedes volksschulrelevante Problem lösen. Wichtig ist es, eine Computersprache zu verwenden, die ohne viel Vorarbeiten (Sprachkenntnisse) rasch den Schüler zum eigenständigen Arbeiten anregt, d.h. er soll schnell dazu kommen, "für ihn selber Bedeutsames" zu tun.

Da nicht die Programmier- sondern die Problemlösefähigkeit im Vordergrund stehen soll, ist eine **interaktive Sprache,** eine, bei der das Problem in einer Art Pseudo-Dialog zwischen Schüler und Computer angegangen werden kann, nötig. - U.a. auch um dem Nimbus entgegenzutreten, eine Geheimwissenschaft zu sein, muss die Programmiersprache deutsche (muttersprachliche) Befehlswörter zur Verfügung stellen. Auch macht nur eine **muttersprachliche Programmiersprache** direkt bewusst, worin die Unterschiede zwischen einer künstlichen und einer natürlichen Sprache bestehen. Die Begriffe einer Computersprache sind sinnleer, semantisch inhaltslos; der Computer arbeitet rein syntaktisch, aber in diesem Bereich völlig eindeutig und sehr kleinlich.

Sprach-Situationen

- Dem Schüler steht eine nicht von ihm selber programmierte Prozedur DREIECK zur Verfügung. Er gibt DREIECK ein und erhält folgende Zeichnung auf dem Schirm:

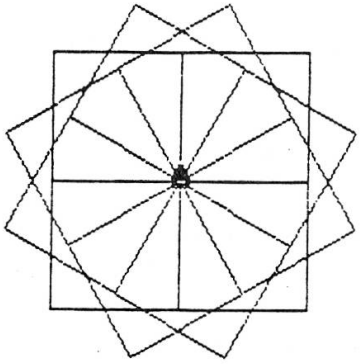


Die Prozedur

```
LERNE DREIECK
WH 6 [VW 50 RE 60]
END
```

trägt ihren Namen also aus semantischer Sicht zu Unrecht, was den Schüler ziemlich irritiert.

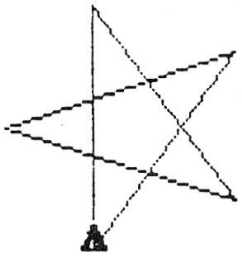
- Zwei Schülerinnen wollen diese Figur erzeugen:



"Wie wollen wir dem sagen?"
"Ja vielleicht ROS4; tipp's ein!"
Dem Computer (also LOGO) ist die Benennung der Programme 'egal'; er kann mit beliebig 'abstrakten' Wörtern wie SISO, NARK, DADU usw. arbeiten; auch orthographisch reagiert er unbedarft: ROSETE, PIRAMIDE, GOESSE usw. sind durchaus legitime, taugliche (Prozedur-)Namen.

- Die Schülerin hat ein Programm STERN entwickelt.

```
LERNE STERN
WH 5 [VW 80 RE 144]
END
```



Sie möchte mehrere Sterne gezeichnet haben und erlebt, dass ihre Eingabe STERNE vom Computer nicht 'verstanden' wird. Der Computer arbeitet nicht nach grammatikalischen Regeln, obwohl er auf muttersprachliche Befehlswörter reagiert.

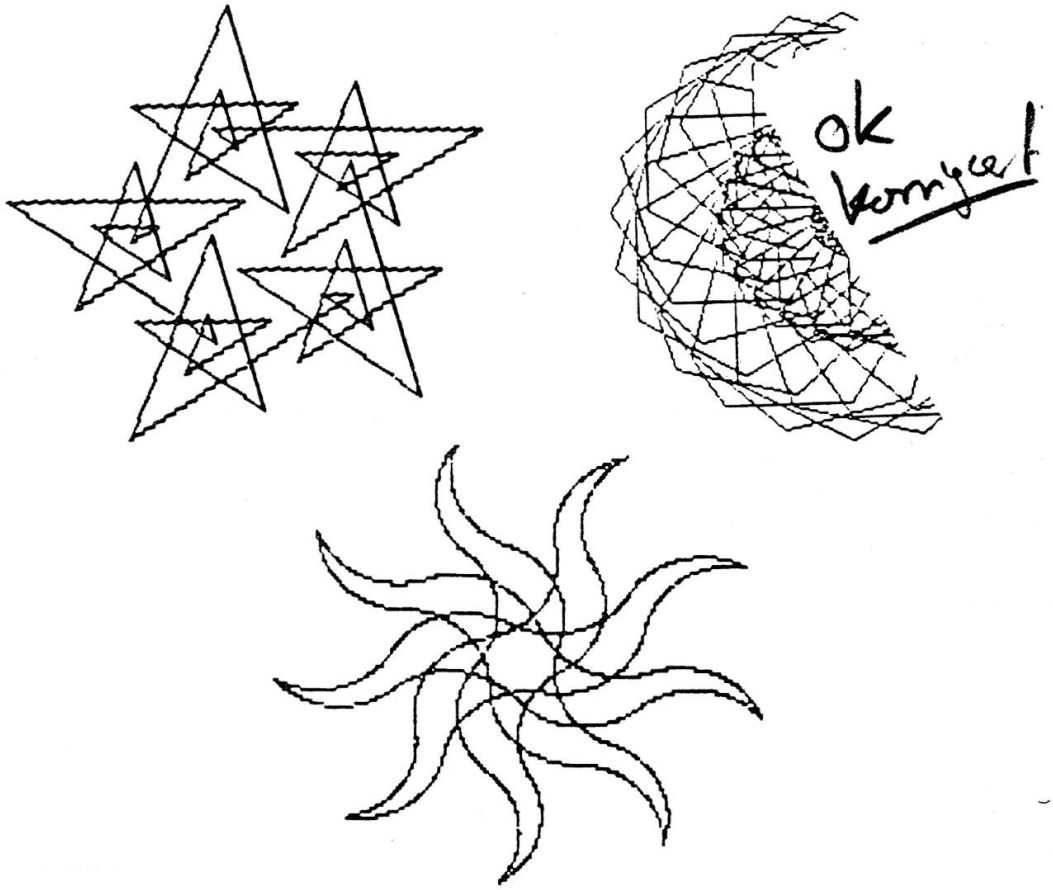
- RECHTS 20
  WIEDERHOLE [DREIECK]

Diese zwei Zeilen gibt ein Schüler ein, in der Meinung, dass der Computer in der neuen Igelrichtung das bereits gezeichnete DREIECK nochmals wiedergebe. Es kommt zu einer Fehlermeldung.
Die Syntax von LOGO verlangt unmittelbar nach WH eine Zahl (Anzahl Wiederholungen); aber mit WH 1 [.....] wird die Befehlsfolge [.....] nur einmal ausgeführt, also eigentlich nicht wiederholt.

Durch die vielen sprachlichen Rand-Erscheinungen und -Erlebnisse, die sich aus dem Spannungsfeld "Muttersprache - Umgangssprache" ergeben, macht der Schüler in dieser Art nebenbei selber Erfahrungen über die Arbeitsweise des Computers, die sonst in einem Einführungskurs in die Informatik meist abstrakt und lediglich verbal vermittelt werden.

Aufgaben, bei denen die Lösungsbemühungen des Schülers immer sofort anzeigen, ob das Ziel erreicht ist oder nicht, sind für die Schule ideal, weil sich die Motivation des Schülers beim Arbeiten eher steigert als verringert. Abstrakte Zahlenaufgaben machen selten oder nie ersichtlich, ob das errechnete Ergebnis auch wirklich stimmt, es sei denn, findige Schulbuchautoren bauen solche Selbstkontrollen ein (Kreuzzahlenrätsel, Rechnen im Stern, Rechenpuzzle usw.). Ganz anders bei gestalterisch-geometrischen Aufgaben: Es braucht keine langen Erklärungen, ob z.B. ein entworfenes Muster wirklich das gesuchte Ornament ist oder nicht. Solche Aufgaben enthalten in sich eine recht grosse motivationale Komponente.

Dieser selbstkorrektive Aspekt des Grafischen und die dadurch entstehende intrinsische Motivation kommen in LOGO voll zum Tragen: Der übliche Einstieg in LOGO wird über die Igelgrafik gemacht, was LOGO vielenorts das "negative" Image einer Kinder-Programmiersprache eingebracht hat, mit der man allenfalls lediglich auf einfache Art ein paar hübsche Grafiken erzeugen könne.



Das ist eine Fehleinschätzung: LOGO ist eine vollausgewachsene Programmiersprache, mit der man durchaus auch auf Mittelschulniveau auf keine engen Grenze stossen wird.

Was LOGO aber gegenüber anderen Sprachen auszeichnet, ist ihr einfacher, problemloser Zugang über die Grafik und ihr durchdachter, das strukturierte Denken unterstützender Aufbau.

Was charakterisiert LOGO als didaktische Programmiersprache?

Der muttersprachliche Befehlswortschatz und die leichte Zugänglichkeit über die Igelgrafik machen LOGO zu einer sehr schülergerechten Programmiersprache. Ausserdem werden, im Gegensatz zu hie und da geäusserten Befürchtungen, die sonstigen schulischen Bemühungen durch den Einsatz von LOGO echt bereichert. Die Igelgrafik stellt einen anderen und möglicherweise für viele Schüler/innen sehr angenehmen Zugang zur Geometrie dar. Soll er allerdings fruchtbar werden, dürfen nicht nur LOGO-Übungen am Bildschirm stattfinden. Ganz nach Bruner's Theorie kann Igelgeometrie enaktiv beginnen: Der Schüler selber "spielt" Igel und vollzieht körperlich die notwendigen LOGO-Befehle, um z.B. ein Dreieck, ein Quadrat, ein reguläres Fünfeck zu "begehen", "abzuschreiten" usw. - letzteres ist überigens mit Volksschülern ohne Igelgeometrie nur durch eine auf diesem Verständnisniveau uneinsichtige Konstruktion zu gewinnen! Damit sich die n-Ecke schliessen, sind notwendigerweise Winkel- und Streckenmessungen nötig.

In Igel-manier u.U. mit Hilfe von Rasterpapier geometrisch-ornamentale Zeichnungen entwerfen, kann sich in einer nächsten Phase anschliessen. Viele der neueren Geometrie-Lehrmittel z.B. für die Mittelstufe enthalten bereits ganz ähnliche Arbeitsblätter und Aufgaben.

Auf diese Weise wird der Schritt über das Ikonische zum Symbolischen auf dem Bildschirm, wo der Igel (der Zeichenstift) nur noch indirekt dirigiert wird, leicht fallen. Analogien zur Steuerung von Robotern, von ferngesteuerten Autos, wie sie jede Spielwarenabteilung eines Warenhauses anbietet, drängen sich unweigerlich auf.

In der Tat arbeiten die Erfinder von LOGO unter Leitung des Piaget-Schülers Seymour Papert am MIT (USA) auch mit einem physischen Igel, einer Plexiglashalbkugel mit Zeichenstift und Motor auf Rädern.

LOGO-Geometrisches

reguläres N.ECK N	Winkel, um den in der Ecke gedreht wird	Summe aller Drehwinkel
3	------	-------
4	90	360
5	------	-------
6	------	-------
7	------	-------
:	:	:
:	:	:
n	------	-------

Werden die Werte für obige Tabelle durch Probieren eruiert, ergibt sich bald der Aussenwinkelsatz für (beliebige konvexe) n-Ecke:

Die Summe aller Aussenwinkel in einem n-Eck beträgt 360°.

Dieser Satz ist ein Spezialfall des allgemeineren (topologischen) Satzes, der besagt, dass die Gesamtdrehung bei einer geschlossenen Kurve in der Ebene ein Vielfaches von 360° ist. In diesem Zusammenhang kann die Prozedur

```
LERNE AHA :SEITENLÄNGE :WINKEL :ZAHL
WH :ZAHL [VW :SEITENLÄNGE LI :WINKEL]
END
```

ausprobiert werden z.B. mit den Werten 144, 150, 216, 32, 72, 96 usw. für :WINKEL.
Wann schliesst sich der Linienzug und wie muss dazu der Wert für :ZAHL lauten?

Die Sichtweise in der herkömmlichen (euklidischen) Schulgeometrie ist eine "von oben"; der Geometriebetreibende ist ein Kartograph der Geometrielandschaft. Hingegen ist der LOGO-Igel, in den sich der Schüler versetzen kann, eine Art Wanderer, der nur lokal die "Übersicht" hat. Durch Wiederhole 360 [VW 1 RE 1] zeichnet der Igel einen Kreis, ohne allerdings "Bescheid zu wissen", wo sich dessen Mittelpunkt befindet.

Für Wertzuweisungen an Variablen (Speicherplätze) verwenden viele Programmiersprachen das Gleichheitszeichen:

z.B. BASIC A = A + 1 oder auch LET A = A + 1
PASCAL A := A + 1

In der (neuen) Mathematik wird von der ersten Primarschulklasse an Wert darauf gelegt, dass die Schüler/innen das =-Zeichen richtig interpretieren und anwenden, nämlich als Zeichen dafür, dass links und rechts davon das Gleiche steht, allenfalls in je anderer Notation.

Nur die LOGO-Schreibweise ist hier konsequent und verwendet das = -Zeichen nicht unangebracht: SETZE "A : A +1.

Auch wird in LOGO sehr genau und streng unterschieden zwischen Variablenbezeichung ("A) und Variableninhalt (:A) - eine nicht zu unterschätzende Denkhilfe für das richtige Verstehen der Funktion von Variablen in der Algebra.

Noch in anderen Bereichen kann LOGO Unterstützung für richtiges Verständnis mathematischer "Instrumente" bieten:

- Der Funktions-Begriff ist neben dem Begriff der Menge einer der tragendsten der (neuen) Mathematik; bereits in der Unterstufe wird mit sog. Maschinen z.B. 2, + 15 gearbeitet. Beim Bruchrechnen, den Dreisätzen, den Proportionen usw. eröffnen diese Maschinen, später Operatoren genannt, neue, andere didaktische Zugänge zu den Themen.

  LOGO stellt mit dem Konzept der lokalen Variablen (hinter dem Prozedurnamen aufgeführt) und dem Rückgabe-Befehl eine adäquate technische Realisierung dieser mathematischen Maschinen zur Verfügung und unterstützt dadurch funktionales, prozedurales Arbeiten.

- "Reihen und Folgen" ist ebenfalls ein Thema, das seinen Platz in der (neuen) Mathematik der Volksschule gefunden hat. Dabei hat man festgestellt, dass die rekursiven Gesetzmässigkeiten in Zahlenfolgen von den Schülern viel schneller und leichter gefunden werden (können) als die funktionalen: z.B. hat die berühmte Fibonacci-Zahlenfolge

  1 1 2 3 5 8 13 21 .....

  das rekursive Bildungsgesetz $a_i = a_{i-1} + a_{i-2}$

  für das i.te Folgenglied.

  Das i.te Glied funktional aus i zu bestimmen ist eine sehr anspruchsvolle Aufgabe.

  Auf einer allerdings abstrakteren Ebene ist die Rekursion, der rekursive Prozeduraufruf ein programmiertechnisch sehr mächtiges Gestaltungswerkzeug in LOGO.

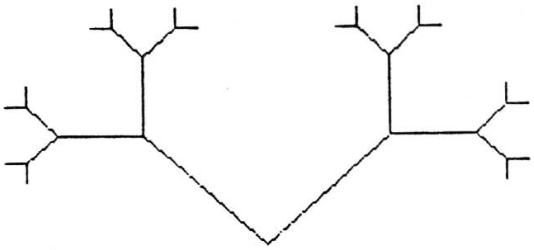
LOGO-Mathematisches

```
LERNE ABS :ZAHL
RÜCKGABE WENN :ZAHL < 0 [OHNEERSTES :ZAHL][:ZAHL]
END

LERNE MASCHINE
RÜCKGABE LIESWORT * 15
END
```

Mit dem Aufruf von ABS muss auch gleich eine konkrete Zahl für die lokale Variable :ZAHL miteingegeben werden. ABS gibt den Absolutbetrag der Eingabezahl zurück.
Wird MASCHINE aufgerufen, so 'erwartet' die Prozedur durch LIESWORT eine Eingabe(zahl) und gibt anschliessend das 15fache davon aus.
Beide Prozeduren entsprechen dem mathematik-didaktischen Vorstellungsmodell 'Maschine' (Operator).

```
LERNE BAUM :LÄNGE
LI 45 VW :LÄNGE
BAUM :LÄNGE / 2
RW :LÄNGE RE 90 VW :LÄNGE
BAUM :LÄNGE / 2
RW :LÄNGE LI 45
END
```



BAUM ist ein typisches und bereits recht komplexes Beispiel für Rekursion:
BAUM 'ruft sich selber' zweimal auf mit je der halben :LÄNGE.

Dadurch, dass LOGO in einfacher Weise ermöglicht, Werte von einer Prozedur zu einer anderen weiterzuleiten, zu übergeben, erlaubt es sogenannt "modulares" Programmieren. Ein grösseres Problem kann in kleinere aufgespalten und diese einzeln gelöst und ausgetestet werden. Dadurch wird eine Problemlösestrategie unterstützt, die nicht nur in der Mathematik ihre Gültigkeit hat, sondern oft sinnvoll einzusetzen ist.

Was darf von LOGO erwartet werden?

Unterscheidet man im ganzen Komplex der Informatikbemühungen in der Volksschule die drei Bereiche

- Lernen *über* den Computer, worunter Themen wie Aufbau und Funktionsweise eines Computers, Datenschutz, informationell geprägte Berufe usw. zu finden sind,
- Lernen *durch* den Computer, also der CUU, worin der Computer als Lerngerät Verwendung findet,
- Lernen *mit* dem Computer, wo der Computer als Hilfsmittel für das Lernen eingesetzt und benützt wird,

so hat LOGO seinen Platz ganz eindeutig im letzten Bereich. Ist man davon überzeugt, dass neben dem Arbeiten mit Anwendersoftware auch exemplarisch das Programmieren in die Informatik der Volksschule gehört, so bietet sich LOGO als *die* "vernünftige" Lösung an.

Nötig ist dabei aber eine für unsere Schule abgestimmte Einsatzdoktrin; Anwendungs- und Verwendungsmöglichkeiten müssen aufgezeigt werden, wobei es nicht darum gehen kann, herkömmliche Stoffe zu verdrängen, aber sie mindestens zu überdenken.

LOGO ist kein Elixier für das schulische Seelenheil und kein Patentrezept gegen alle, insbesondere die motivationalen Probleme im Zusammenhang mit dem Mathematikunterricht. Sowas gibt es wohl nie, auch wenn es immer wieder versprochen wird.

Immerhin, LOGO ist anregend und lässt viel Raum für Eigenaktivitäten der Schüler. Die Zukunftsperspektiven von LOGO hängen stark davon ab, ob den Schülern ein Freiraum dieser Art gestattet werden soll oder nicht.

LITERATUR

Feuerstein, G.: *LOGO und die Freude an der Selbstdisziplin* in: mathematiklehren, Heft 9 / April 85.

Hoppe, H.U.: *LOGO im Mathematikunterricht.* IWT Verlag, Vaterstetten, 1984.

Hoppe, H.U.: *Anforderungen an Progammiersprachen für den Unterricht unter dem Gesichtspunkt des Interaktiven Progammierens.* Aus: Mandl/Fischer (Hrsg.), Lernen im Dialog mit dem Computer, Urban & Schwarzenberg, München, 1985.

Löthe, H.: *Rechnereinsatz in der Primarschule und der Sekundarstufe I.* In: die neue schulpraxis, Heft 11 / 85.
Papert, S.: *Mindstorms, Kinder, Computer und neues Lernen.* Birkhäuser Verlag, Basel, 1982.
Watt, D.: *LOGO - Computersprache für Eltern und Kinder.* te-wi Verlag, München, 1984.

RESUME

Aspects du LOGO

Quelle est la fonction de l'ordinateur à l'école? L'enseignement assisté par ordinateur (computer assisted instruction), le software applicable et la programmation, trois domaines importants pour l'école, sont discutés.

Pourquoi apprendre à programmer? Par le biais de la programmation, l'ordinateur devient un «outil de pensée», et cette fonction ne peut pas être atteinte par la seule application du software.

Pourquoi choisir LOGO? Ce langage est orienté vers le dialogue en utilisant des commandes dans la langue maternelle et il simplifie les tâches de l'élève grâce à son excellente présentation graphique.

En quoi LOGO est-il un langage de programmation didactique? La «géométrie de la tortue» peut permettre une nouvelle approche de la géométrie. De plus, LOGO propose des aides visuelles pour faire comprendre des notions de mathématiques proprement dites.

Que peut-on attendre de LOGO? LOGO n'est certainement pas le remède universel contre tout problème scolaire. Mais LOGO est motivant et c'est l'un des instruments les plus utiles pour l'enseignement de l'informatique à l'école.

SUMMARY

Aspects of LOGO

What functions do computers have in school? Computer assisted instruction, software application and programming are three such fields.

A reason for programming nevertheless: Through programming the computer becomes a "think-instrument", this aspect can not be covered by only applicating software.

Why choose LOGO? LOGO is a dialog-oriented language with colloquial commands and it offers easy access through its exellent graphic capabilities.

What makes LOGO a didactic programming language? Turtle geometry offers quite a new access to geometry and apart from this, LOGO supplies help for the correct understanding of many mathematical concepts.

What can be expected from LOGO? LOGO can not be the universal remedy for all scolastic problems. But LOGO is motivating and one of the most sensible and helpful instruments for teaching informatics in public schooling.