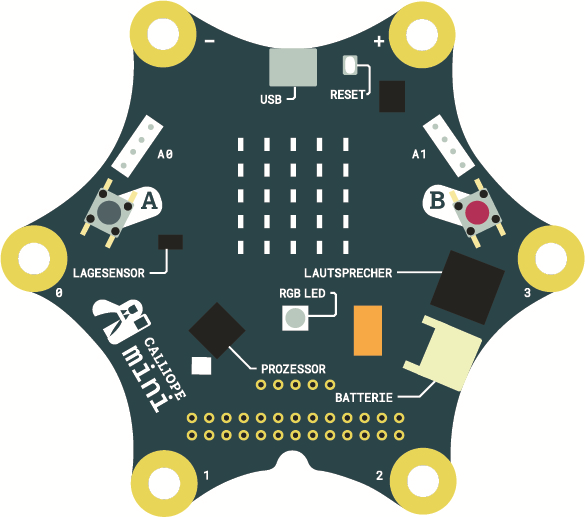


**Calliope mini**



Quelle: https://calliope.cc/content/2-idee/2-ueber-mini/calliope\_mini\_grafik.png

**17. Oktober 2018**Mittelschule Ingolstadt-Oberhaunstad

Bernd-Rosemeyer-Str. 1

85055 Ingolstadt

**Quellennachweis:**

Alle hier verwendeten Grafiken sind Grafiken der Interseite: calliope.cc oder Screenshots des jeweiligen Editors.

Quellen für weitere Grafiken sind zusätzlich angegeben.

Willibald Heßlinger ⬩ Mittelschule Ingolstadt-Oberhaunstadt⬩ Mail: willi.hesslinger@web.de

# Vorbemerkung

Der **Calliope mini** ist ein Einplatinencomputer, der für Bildungszwecke entwickelt wurde und an deutschen Grundschulen eingesetzt werden soll. Ziel ist es, alle Schülerinnen und Schüler ab der dritten Klasse mit einem Calliope mini auszustatten.  
Seit August 2017 ist der Calliope mini über den Cornelsen Verlag und die Versandhändler Exp-Tech, Conrad Electronics SE und Amazon erhältlich. Der Einzelpreis beträgt etwa 35 €, darüber hinaus sind bei Cornelsen auch Klassensätze erhältlich.

Schaltpläne und Software des Calliope mini sind als **Open Hardware** beziehungsweise **Open Source** frei verfügbar, und es stehen mehrere Anleitungen als **Open Educational Resources** zur Verfügung.

Dem Calliope mini wurde 2016 der Wolfgang Heilmann-Preis für humane Nutzung der Informationstechnologie verliehen. Ebenso erhielt das Bildungsprojekt den "The Innovation in Politics Awards 2017" und den "Innovationspreis für digitale Bildung delina 2018" in der Kategorie Frühkindliche Bildung und Schule. (Quelle: Wikipedia)

Der Name stammt von **Kalliope, einer der Töchter des griech. Gott Zeus**: Muse der epischen Dichtung, der Wissenschaft, der Philosophie und des Saitenspiels

# Material zum Calliope mini

Mittlerweile gibt es sehr viel und vor allem gutes Material für die Handhabung und die die unterrichtliche Umsetzung des Calliope mini. Es ist eher das Problem, dass man aus der Fülle des Angebots (größtenteils kostenlos) das geeignete Material findet.

|  |  |
| --- | --- |
| [**https://calliope.cc/**](https://calliope.cc/) | **Hauptseite** der Calliope GmbH, 10437 Berlin    Kostenloser Download von Arbeitsheften und sonstigem Material für den Unterricht!! |
| [**www.cornelsen.de/calliope/**](http://www.cornelsen.de/calliope/) | Seite des Cornelsen-Verlags über Material und Bestellung des Calliope mini |
| [**https://tuduu.org/lernen/calliope**](https://tuduu.org/lernen/calliope) | Junge Tüftler – tolle Seite mit Videos |
| **youtube** | Bitte Suchbegriff „Calliope mini“ eingeben. |

**TIPP:**

|  |
| --- |
| [**https://calliope.cc/schulen/schulmaterial**](https://calliope.cc/schulen/schulmaterial) |
| Hier finden Sie eine Auflistung der besten Projektseiten mit Unterrichtsmaterial |

# Erster Kontakt mit dem Calliope mini

* Schließen Sie den Calliope über das USB-Kabel an den Computer an.
* Beobachten Sie, was dabei geschieht.
* Drücken Sie die Taste „A“ und/oder „B“.
* Schütteln Sie den Calliope

Beobachten Sie immer was auf der LED-Anzeige des Calliope passiert.

# Die Bestandteile des Calliope

**Hinweis**:

Falls Sie später mit einem Tablet arbeiten, gelangen Sie durch **Einscannen der QR-Codes** automatisch auf die angegebene Internetseite.

Die Bestandteile des Calliope mini sind auf folgender Internetseite gut erklärt.

|  |  |
| --- | --- |
| QR-Bestandteile | [**https://calliope.cc/idee/ueber-mini**](https://calliope.cc/idee/ueber-mini) |
| [**https://tuduu.org/lernen/calliope**](https://tuduu.org/lernen/calliope) |  |

Weitere Informationen erhalten Sie auf der **Datei/Arbeitsblatt:**



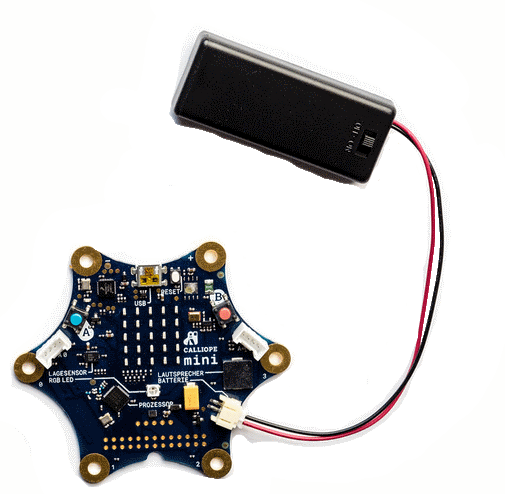
# Inbetriebnahme des Calliope mini

* **Betrieb am Computer:**Schließen Sie einfach den Calliope über den USB-Anschluss.

|  |  |
| --- | --- |
|  | Im Explorer erscheint der Calliope als eigenes Laufwerk |
|  | Durch Klicken auf die Datei „MINI.HTM“ gelangt man zur Hauptseite von Calliope: **calliope.cc** |

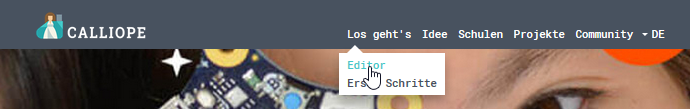
* Der Calliope piept kurz und wird vom Computer erkannt.

Hier wird nun der **Editor** ausgewählt. Die Editoren werden in nächsten Schritt genauer betrachtet.



* **Betrieb ohne Computer**Mit dem mitgelieferten Batterieblock und den beiden AAA-Batterien kann der Calliope ebenfalls betrieben werden.  
  Dazu muss das kleine, weiße Plastikende (JST-Stecker) vorsichtig in den auf dem Calliope mini befestigten Batterieanschluss gesteckt werden.   
  Achten darauf, dass die geschlossene Steckerseite mit der Erhöhung in der Mitte oben ist. Wenn der Kabelstecker leicht in den Batterieanschluss des Boards passt, ist es richtig.
* **Betrieb mit dem Tablet**Das Calliope mini kann über Bluetooth mit dem Tablet angesteuert werden

# Die Editoren für den Calliope mini

****

Nach dem Aufruf der Calliope-Homepage können Sie im Menu „Editor“ zwischen **3 verschiedenen Editoren** auswählen.

|  |  |
| --- | --- |
| calliope_mini | Calliope mini Editor Die einfachste und schnellste Art den mini zu programmiren [**miniedit.calliope.cc**](https://miniedit.calliope.cc/) |
| logo-inverted | MakeCodeProgrammieren durch das Anordnen von Programmier-Blöcken oder mit Javascript. [**makecode.calliope.cc**](https://makecode.calliope.cc) |
| logo_open_roberta_rgb | Open Roberta Lab®Programmieren mit grafischer Oberfläche wie bei den obigen Beispielen. Die Open-Source-Plattform beruht auf der langjährigen Erfahrung der Fraunhofer-Initiative „Roberta® – Lernen mit Robotern“. [lab.open-roberta.org](https://lab.open-roberta.org) |



Eine Gegenüberstellung der Editoren erhalten Sie mit Hilfe der Datei:

# Erstes Programm mit dem Calliope mini

**Hinweis:**

Es besteht oft die Gefahr, dass man nur ein Programm auf das andere Programm folgen lässt und die Schnellen machen dann noch schnell die nächste Stufe und so weiter.

**Bitte achten Sie darauf, dass Sie immer die einzelnen Programmierbeispiele zusammen mit den Schülern besprechen/reflektieren und den Algorithmus reflektieren.**

**Dies entspricht nicht nur dem kompetenzorientierten Unterricht, sondern zeigt auch den Schülern auf, dass sie hier nicht nur „spielen“, sondern wie richtige Programmierer Abläufe kreieren.**

**Vorgehensweise:**

Für den Vergleich der einzelnen Editoren wird mit den verschiedenen Editoren das gleiche Programm erstellt.

Danach werden die Ergebnisse in die Tabelle desArbeitsblattes**:**eingetragen.

Dieser Vergleich der Editoren kann auch im Unterricht übersprungen werden. Sie können gleich mit dem Editor Ihrer Wahl die ersten Programmierübungen mit Ihren Schülern starten.

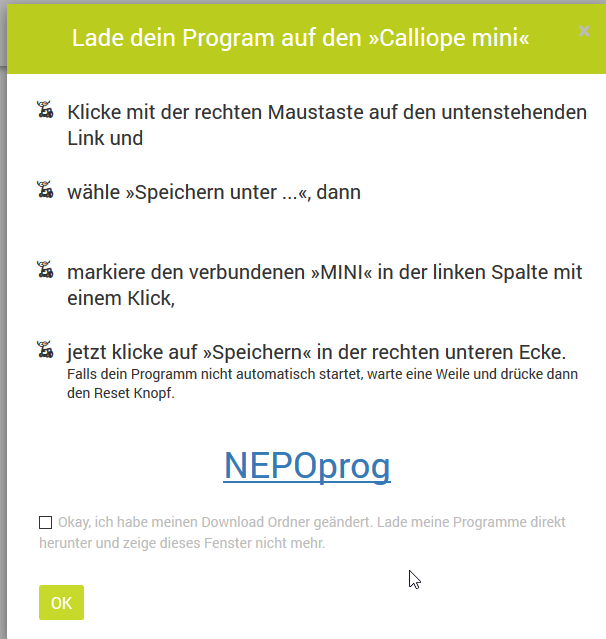


Der Ablauf des ersten Programms wird auf dem Arbeitsblatt:

beschrieben. Es wird mit den verschiedenen Editoren ein „Smiley“ auf dem Calliope erzeugt.

# Übertragen des Programms auf den Calliope mini

Damit das erstellte Programm auch auf dem Calliope mini ausgeführt werden kann, muss dieses auf **den Minicomputer übertragen** werden.

Die Übertragung unterscheidet sich je nach verwendetem Editor.

|  |  |
| --- | --- |
| calliope_mini | Calliope mini Editor Auf „**Herunterladen**“ klicken und dann per **„drag & drop“** auf das Laufwerkssymbol ziehen. |
| logo-inverted | MakeCode Auf „**Herunterladen**“ klicken, das Programm wird in den Ordner „Download“ kopiert  oder über das **Diskettensymbol** den Speicherort wählen. |
| logo_open_roberta_rgb | Nepo - Open Roberta Lab® Auf das Icon „Starte auf Calliope“  Klicken und im anschließenden  Dialogfenster den Speicherort  Wählen |

# Links und Tipps für Material

* **Calliope Projektseite** mit Unterrichtsbeispielen  
  <https://calliope.cc/>
* **Calliope Forum**  
  <https://forum.calliope.cc/>
* **Projektideen**  
  <https://calliope.cc/projekte>
* **Tuduu**   
  (Plattform für Kids, Teens und Erwachsene, die basteln, bauen, tüfteln wollen)  
  <https://tuduu.org/lernen/calliope>
* Calliope im Saarland

[**http://calliope-saarland.de/category/unterrichtsmaterial/**](http://calliope-saarland.de/category/unterrichtsmaterial/)

**mit Lernkarten, die auf dem Open Roberta Editor aufbauen**Das Land Saarland stellt den Schulen einen Satz Calliope kostenlos zur Verfügung, wenn eine Lehrkraft, die angebotenen Kurse absolviert hat.

* **Landesinstitut für Pädagogik und Medien - Saarland**[**https://www.lpm.uni-sb.de/typo3/index.php?id=5893**](https://www.lpm.uni-sb.de/typo3/index.php?id=5893) **(Tolles Material und Videoanleitungen)**
* **Kika-Beitrag zu Calliope**  
  <https://www.kika.de/timster/videos/video45586.html>
* Seite eines **Gymnasiallehrers**   
  <https://calliopemini.info>
* **App Camps**(Informationen zum Material  
  <https://appcamps.de/unterrichtsmaterial/calliope-mini/>
* **Materialseite von** **AppCamps** (nach vorheriger Registrierung – nicht kommerziell)  
  <https://teach.appcamps.de>

# Übungen zu den beiden Editoren „Calliope mini“ und „Open Roberta“

## Material des Cornelsen Verlags: Nepo – Open Roberta Lab

Je nach Wahl des Editors kann für die erste Begegnung das Konzept des Cornelsen Verlags übernommen werden. Es können hierbei die Arbeitshefte, die bei der Bestellung eines Klassensatzes kostenlos mitgeliefert werden, verwendet werden oder Sie laden sich das Arbeitsheft im pdf-Format unter folgender Internetadresse auf Ihren Rechner:

|  |  |
| --- | --- |
|  | **Schülerarbeitsheft** ab 3. Klasse auch für Klassen der MS geeignet.  <https://calliope.cc/schulen/schulmaterial> |
|  | **Lehrerarbeitsheft** ab 3. Klasse auch für Klassen der MS geeignet.  <https://calliope.cc/schulen/schulmaterial> |

## Lernkarten der Saarland-Seite

[**http://calliope-saarland.de/category/unterrichtsmaterial/**](http://calliope-saarland.de/category/unterrichtsmaterial/)

## Weitere Lernkarten

Weitere Lernkarten erhalten Sie über die Eingabe des Suchbegriffs „Lernkarte Calliope“ im Browser Ihres Computers.

**ANHANG - PROGRAMMIEREN**

# Vorbemerkung

# Egal welche Programmieranweisungen man im Unterricht erstellt oder welche Programmiersprache/Editor man verwendet, sollte man die Schüler dazu anhalten, dass Sie für den Programmablauf reflektieren. D. h. es geht nicht darum, dass eine Programmieranweisung nach der anderen erstellt wird, sondern den Schülern sollte die jeweilige Struktur und die Art des Ablaufs für ein Programm veranschaulicht werden. Hierzu können Diagramme oder Organigramme sehr nützlich sein.

# Information

Für die erste Begegnung der Schüler mit Programmierinhalten bietet sich die Möglichkeit an, dass diese in Partnerarbeit die Rolle von „Roboter“ und „Programmierer“ einnehmen. So wird der Bereich Programmieren zunächst aktiv handelnd erlebt, ebenso wird hiermit das Verständnis für klare und folgerichtige Anweisungen geschaffen. Weiterhin kann in dieser Phase einmal die „Rolle“ des Roboters und des Programmierers eingenommen werden und so wiederum das Verständnis für spätere Algorithmen angebahnt werden.

Ebenso sollte an dieser Stelle auch schon die Reflexion und Versprachlichung der Aufgaben eingeführt werden.

# Rahmenbedingungen

Je nach Art der örtlichen Begebenheiten und der Klassensituation kann diese Einführung auf zwei verschiedene Arten erfolgen.

## Programmanweisungen im Raum(Parcours)

**Parcours** im Klassenzimmer, in einem Fachraum, auf dem Pausenhof, usw., in dem die Schüler selbst die Akteure sind und sich dort nach Programmieranweisungen bewegen.

## Programmanweisungen im Rastersystem (Arbeitsblatt)

**In einem Rastersystem** auf einem Arbeitsblatt können die Schüler selbst handelnd Objekte bewegen und die Handlungsanweisungen (Programmierbefehle) formulieren und diskutieren.

Der Lehrkraft ist es selbst überlassen, ob diese beide Zugänge für die ersten Programmversuche durchläuft oder sich nur auf einen der beiden Einführungssituationen beschränkt.

## Hinweise zu den ersten Programmieranweisungen

**Wichtig** ist, dass neben dieser Handlungsebene bei den Schülern auch die Erkenntnis geschaffen werden, dass der **„Roboter“ (Maschine) keinen Verstand hat** und bei jeder Aufgabe eben „Schritt für Schritt“ angeleitet werden muss.

Wenn der Roboter keinen Strom hat (ausgeschaltet wird), dann kann er nichts machen, sehen, fühlen, usw.

Die Sensibilisierung der Schüler kann mit folgenden Impulsen unterstützt werden:

* „Welche Fähigkeiten besitzt ein Roboter, wenn dieser keinen Strom mehr hat?“
  + „Denke an dein Smartphone, wenn der Akku leer ist!“
* Was muss der Roboter beim Einschalten alles lernen?“
  + „Beim Einschalten deines Smartphones muss dieses alle Funktionen neu laden (lernen)“
* „Für jede Tätigkeit benötigt der Roboter eine Anleitung“
  + „Damit du mit deinem Smartphone ein bestimmtes Spiel „zocken“ kannst, musst du erst das Spiel (App) auf dem Smartphone laden.“
* „Für jede Fähigkeit benötigt der Roboter ein Zusatzgerät“
  + „Je nach Ausstattung „beherrscht“ dein Smartphone bestimmte Dinge: Es kann gut Fotos machen (je nach Art der Kamera), es kann Musik gut widergeben (je nach Art des Lautsprechers).
  + Der Roboter kann nur sehen, wenn …
  + Der Roboter kann nur Sound ausgeben, wenn ….
  + Der Roboter kann nur „hören“, wenn …

# Vorschlag für eine Einführungssequenz

## Programmanweisungen im Raum(Parcours)

**Vorbereitung**Es sollte zunächst ein Parcours im Klassenzimmer, Fachraum, Pausenhof, Sporthalle, … aufgebaut werden. Dies kann mit Tischen, Bänken, Stühlen, Klebebändern auf dem Boden, … geschehen.

**Aufgabe**

1. Ein Kind spielt den Roboter, die Lehrperson gibt Anweisungen, welche das Kind sofort ausführt. Die Aufgabe der anderen Kinder ist es, sich möglichst alle Anweisungen zu merken.

**a.** Startsignal: Leichte Berührung am Nacken

**b.** 1 Schritt vorwärts: Berührung auf den Rücken

**c.** ¼ - Drehung nach rechts: Berührung der rechten Schulter

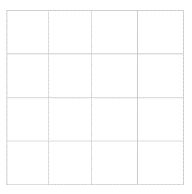
**d.** ¼ - Drehung nach links: Berührung der linken Schulter

**e.** Stoppsignal: Leichte Berührung am Nacken

1. Nach dieser Übung wird besprochen, welche Befehle sich die Kinder merken konnten und was beachtet werden muss, wenn das Spiel nun in Partnerarbeit durchgeführt wird. Es darf nichts gesprochen werden.
2. Das gleiche Spiel wird nun in Partnerarbeit durchgeführt. Die Lehrperson gibt ein Zeichen für den Wechsel.
3. In der Gruppe wird besprochen, was (nicht) gut funktioniert hat und weshalb.
4. In Partnerarbeit wird noch einmal das gleiche Spiel durchgeführt. Nun kann der Programmierer allerdings vier (oder später mehr) Anweisungen auf einmal programmieren und dies mit einer Berührung beider Schultern gleichzeitig speichern. Erst nach dem Startsignal läuft das Kind die programmierte Strecke ab.

**Hinweise zur Durchführung**

Es ist wichtig, dass die Kinder während der Aufgabe verstehen, was sie machen. Deshalb sollte großen Wert auf die Diskussion (Punkte 2 und 4) gelegt werden.   
In der zweiten Partnerarbeit (Punkt 5) kann es hilfreich sein, wenn die Lehrperson die Aufgabe noch einmal mit einem Kind vorzeigt. Zur Differenzierung/Vertiefung kann die Lehrperson den stärkeren Schülerinnen und Schülern den Auftrag geben, 6 oder 8 Anweisungen zu speichern.   
Alternativ können die Kinder mit vier Anweisungen starten und jedes Mal, wenn der “Roboter“ die programmierte Strecke korrekt abgelaufen ist, die Anzahl Anweisungen um eins erhöhen. Ansonsten überschätzen sich einige Kinder möglicherweise und programmieren so viele Anweisungen, dass der Roboter irgendeine Strecke läuft und der Programmierer auch nicht mehr so genau weiß, was er nun eigentlich programmiert hat.

**Weitere Aufgabenstellungen**

Die Lehrkraft teilt den Parcours in Quadrate ein, die jeweils einem Schritt (oder 2 Schritten) entsprechen. Nun können die Anweisungen

**ZIEL**

für das Erreichen des Ziels auf dem Rücken programmiert werden

und evtl. auch in Papierform (dies darf der Roboter nicht sehen) notiert werden.

**START**

# Reflexion der Programmierversuche

# Zur Erarbeitung und Sicherung der durchgeführten Unterrichtseinheiten ist es wichtig, dass die Schüler die Anweisungen (Programmierbefehle) reflektieren. Hier sind evtl. Wortkarten oder Impulse zur Versprachlichung wichtig.

„Die Arbeitsanweisung war für mich nicht eindeutig, da ….“

„Die Arbeitsanweisung war für mich eindeutig, da …“

„Der Roboter müsste noch folgende Eigenschaften haben, damit ….“

„Um das Ziel zu erreichen, müsste ….“

„Man könnte die Befehle vereinfachen, wenn …“

„Folgende Befehle müssten noch ergänzt werden: ….“

# Notieren der Programmstrukturen

# Um später auch umfangreichere Programmabläufe zu dokumentieren, sollte man schon zu Beginn einheitliche Symbole einführen. Diese graphischen Hilfsmittel sind für die Logik eines Programmablaufes wichtig, um Abläufe und Verzweigungen anschaulich darzustellen.

# Ebenso erleichtern diese Symbole diese Symbole den späteren Zugang zum Programm „scratch“, das auf diesen Strukturen aufbaut.

# Folgende graphische Elemente werden zunächst verwendet:

|  |  |
| --- | --- |
| Symbol | Bedeutung |
| Oval | START und ENDE |
| Rechteck | Befehl, Operation |
| Pfeil, Linie | Verbindung zu nächsten Element |

# Beispiel:

Rücken

linke Schulter

Rücken

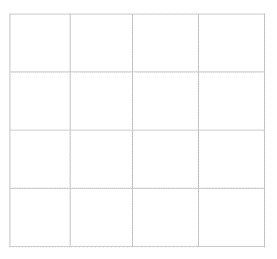
Nacken

Rücken

Rücken

Nacken

# Für komplexere Anweisungen können diese Symbole noch erweitert werden.



**START**

**ZIEL**

|  |  |
| --- | --- |
| Symbol | Bedeutung |
| Raute | Verzweigung, Bedingung |
| Parallelogramm | Ein- und Ausgabe |

## Programmanweisungen im Rastersystem (Arbeitsblatt)

Die ersten Erfahrungen im Parcours werden nun auf das Rastersystem eines Arbeitsblattes übertragen. Auch hier gilt, dass eine Versprachlichung und Reflexion für das „Programmieren“ sehr hilfreich ist.

|  |  |
| --- | --- |
| Mithilfe von Rasterdarstellungen und Grafiken können eindeutige Situationen geschaffen werden.  Die Schüler formulieren dazu passende Anweisungen. Je nach Unterrichtssituation können Lehrkräfte und Schüler selbst Rastersysteme entwickeln und dazu Programmieranweisungen formulieren.  Hier können schon **Bedingungen** und **Schleifen** als Bestandteil einer Programmieranweisung  Eingeführt werden. |  |

Grafiken: alle Grafiken sind Bestandteile der Figurenbibliothek des Programms „Scratch“ - <https://scratch.mit.edu>

# Hinweise zur Erweiterung der Anweisungen

In dieser Einführungsphase soll den Schülern bewusst werden, dass dem „dummen“ Roboter jeder Schritt und jede Bewegung „gelernt“ werden muss. „Wir brauchen (Progammier-)Befehle für den Roboter“.

Diese Befehle können aber erweitert werden. Dazu benötigt der Roboter aber noch verschiedene Reize:

* **Sprache**: Reizaufnahme über das Ohr (Schnittstelle)  
  oder
* **Berührung**: Reizaufnahme durch Sensoren (Tastsinn)

Der Befehl „gehen“ erhält nun eine ganz andere Bedeutung und die Schüler erlangen die Kompetenz, dass hier der Befehl „gehen“ differenzierter betrachtet werden muss.  
z. B. gehe vorwärts, gerne rückwärts, gehe einen Schritt, gehe 2 Schritte, …

Ebenso werden die Schüler schon selbst verschiedene Befehle formulieren, die im Klassenverband vorgestellt und differenziert betrachtet werden.

„Gehe so lange bis …“

Hier kann auch schon die Einführung der Begriffe

* „**Schleife**“ (Wiederholung)  
  und
* „**Bedingung**“ („Wenn …, dann …)

geschehen.

# Ausblick

Mit der bisherigen konkreten Erfahrung der Programmierung kann nun in einem weiteren Schritt eine Programmiersprache bzw. ein Programm hierzu eingeführt werden. Nach der konkreten handelnden Ebene kann nun eine erste Stufe der Abstraktion durchlaufen und das Verständnis für Algorithmen angebahnt werden.

Es bieten sich hierzu – je nach Ausstattung der Schule – folgende zwei Programme an:

* „**SCRATCH**“ als reine objektorientierte Programmiersprache  
  Die Software kann entweder auf einem Rechner installiert werden oder mithilfe eines Browser als online-Version ausgeführt werden.
* „**Calliope**“ als Minicomputer mit eigener Programmiersprache.  
  Das "Calliope mini" ist eine Platine, auf der ein Mikroprozessor und zahlreiche weitere Bauteile angebracht sind (Preis: ca. 35 Euro). Der Vorteil für die Lehrkraft ergibt sich dadurch, dass alle Bauteile enthalten sind und somit nicht verloren werden können. Diese Platine wird über ein USB-Kabel mit dem Rechner verbunden. Es Software kann „Open Roberta Lab“ empfohlen werden.