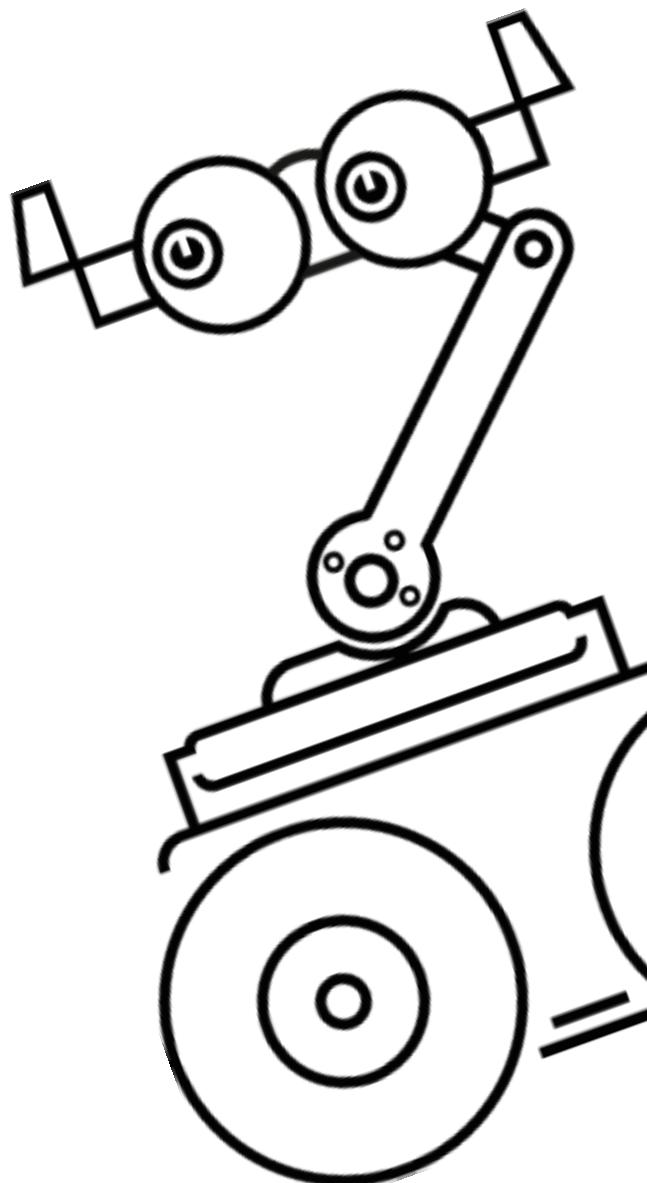


# Programmieren mit Open Roberta

Einführung und Unterrichtsbeispiele mit dem Calliope mini



**ROBERTA**  
INITIATIVE

Roberta ist ein eingetragenes Warenzeichen der Fraunhofer-Gesellschaft e.V.  
Roberta ist seit 2010 Mitglied der Fraunhofer Academy

## Inhalt

Dieses Dokument stellt das zweite einführende Kapitel aus der Reihe [Roberta-Lernmaterialien](#) dar.

Kapitel 2 behandelt die Thematik

# Darum sollten Ihre Schülerinnen und Schüler programmieren lernen!

Die weiteren Kapitel sind:

[Kapitel 1: Programmieren/Coding](#)

[Kapitel 3: Programmieren im Unterricht](#)

[Kapitel 4: Der Calliope mini](#)

[Kapitel 5: So geht Open Roberta!](#)

Unterrichtseinheiten mit Open Roberta und Calliope mini

- Musikstunde
- Primarstufe: Deutsch
- Primarstufe: Mathematik
- Sachunterrichtsstunde: Belebte Natur

# Darum sollten Ihre Schülerinnen und Schüler programmieren lernen!

Computer, Tablets und Smartphones sind in der Lebenswirklichkeit Ihrer Schülerinnen und Schüler allgegenwärtig. Die meisten von ihnen nutzen diese Geräte täglich – oft allerdings, ohne darüber nachzudenken. Was jedoch, wenn die App nicht funktioniert wie gedacht oder das Textverarbeitungsprogramm etwas ganz Anderes macht als gewollt? Viele Nutzer sind in diesen und anderen Fällen ratlos. Dann entsteht schnell der Eindruck, als ob sie es sind, die von der Hard- und Software »programmiert« werden – und nicht umgekehrt.

## »Program or be programmed?«

(Douglas Rushkoff, 2011)

Ihre Schülerinnen und Schüler sollen nicht nur passive Nutzer der digitalen Angebote sein. Sie sollen diese aktiv mitgestalten, Funktionsweisen durchschauen und sich bei Problemen selbst zu helfen wissen. Der Vorteil: Kinder und Jugendliche sind neugierig, vor allem bei neuartigen und unbekannten Dingen; sie wollen lernen, wie ihre Umwelt funktioniert. Dies erleben sie nur, wenn sie selbst tätig werden, erproben und experimentieren. Mit anderen Worten: Wenn sie selbst programmieren lernen.

Programmieren ist dabei jedoch weitaus mehr als nur das Schreiben von Programmen für den Computer. Programmieren erfordert vielmehr Problemlösefähigkeiten, die in allen Bereichen des alltäglichen Lebens und in der Schule notwendig sind. Um komplexe Probleme zu lösen – sei es analog oder digital – braucht es eine Reihe von Denkwerkzeugen. Diese Denkwerkzeuge zum Lösen von Problemen zu nutzen, wird als »Computational Thinking« bezeichnet (Jeannette M. Wing, 2017).

Der Problemlösungsprozess selbst lässt sich durch einen informatischen Modellierungskreislauf (Ludger Humbert, 2006) beschreiben (Abb. 1).

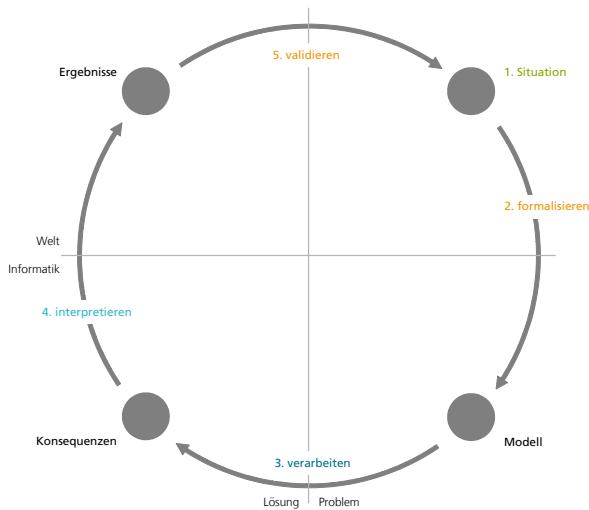


Abbildung 1: Informatischer Modellierungskreislauf (Humbert 2006)

Darin wird zunächst ein Problem beschrieben **1. Problembeschreibung**.

Beispiel in der Programmiersprache NEPO®:

1. Situation: Ich möchte, dass der Calliope Mini »Hallo« anzeigt, wenn ich den Knopf A drücke.

Beispiel in der Alltagssprache:

1. Situation: Ich möchte Schokomuffins backen, weiß aber nicht, wie das geht.

Das Problem muss dann verschriftlicht bzw. in einem Modell formalisiert werden, dessen Zeichen in einer festgelegten Reihenfolge auf festgelegte Art und Weise verarbeitet werden können

## 2. Formalisierung.

Beispiel in der Programmiersprache NEPO®:

2. Formalisierung in Form der Programmiersprache NEPO®:



Beispiel in der Alltagssprache:

2. Formalisierung in Form eines Rezepts.

### Amerikanische Schokomuffins

Menge: 12 Muffins

Vorbereitung: 5 Minuten

Backzeit: 20 Minuten

#### Zutaten:

250 g Mehl

40 g Kakao

100 g Zucker

2 TL Backpulver

1 Prise Salz

100 g Zartbitterschokolade

3 Eier

125 ml Öl

200 ml H-Milch

Schokodrops

#### Zubereitung:

Schmelze die Zartbitterschokolade im heißen Wasserbad. Heize den Ofen auf 180° Ober- und Unterhitze vor.

Vermische alle trockenen Zutaten in einer Schüssel: Mehl, Kakao, Zucker, Backpulver, Salz.

Verquirle anschließend die feuchten Zutaten und menge sie unter die trockenen Zutaten.

Rühre die geschmolzene Zartbitterschokolade unter die fertige Masse.

Füll den Teig in Muffinförmchen und bestreue diese nach Belieben mit Schokodrops.

Backe die Muffins etwa 20-25 Minuten im Ofen.

Das Problem wird in einem nächsten Schritt verarbeitet und gelöst [3. Verarbeitung](#).

Beispiel in der Programmiersprache NEPO®:

3. Verarbeitung und Lösung des Problems:  
Anschließen des Calliope Mini und Ausführen des Programms.

Beispiel in der Alltagssprache:

3. Verarbeitung und Lösung des Problems: Backen der Schokomuffins nach vorgeschrriebenem Rezept.

Das gelöste Problem muss schließlich in der Realität interpretiert werden [4. Interpretation](#).

Beispiel in der Programmiersprache NEPO®:

4. Hat der Calliope Mini »Hallo« angezeigt?

Beispiel in der Alltagssprache:

4. Sind Schokomuffins entstanden?

Die Ergebnisse stellen dann wiederum den Ausgangspunkt zur Überprüfung, Korrektur, Verfeinerung und ggf. Verbesserungen der entwickelten Lösung dar. Es erfolgt die [5. Validierung](#) in der Realität und möglicherweise die Überarbeitung der Lösung zum Zwecke der »Optimierung«.

Beispiel in der Programmiersprache NEPO®:

5. Bewertung der Lösungsrepräsentation:  
Zeigt der Calliope Mini »Hallo« an, wenn Knopf A gedrückt wurde?  
Optimierung:  
Kann die Anweisung verkürzt werden?

Beispiel in der Alltagssprache:

5. Bewertung der Lösungsrepräsentation:  
Schmecken die Schokomuffins?  
Optimierung:  
Was muss am Rezept geändert werden, damit die Schokomuffins besser gelingen?

Die aufgeführten Punkte verdeutlichen die Vielfalt des Programmierens auf verschiedenen Gebieten des Lernens in der Schule. Viele Kompetenzen werden geschult und entwickelt, denn Coding findet sich in vielen Bereichen des Lehrplans wieder.

# Kontakt

Die Roberta-Initiative im Web

[roberta-home.de](http://roberta-home.de)

[lab.open-roberta.org](http://lab.open-roberta.org)

FAQ rund um die Roberta-Initiative

[roberta-home.de/faq](http://roberta-home.de/faq)

Informationen zum Datenschutz

[roberta-home.de/datenschutz](http://roberta-home.de/datenschutz)



## Info

Dieses Material wurde zusammen mit Prof. Dr. Julia Knopf und Prof. Dr. Silke Ladel entwickelt.

Dieses Material entstand mit Unterstützung der Google Zukunftswerkstatt.

Lizenz: CC-BY-SA 4.0

Version: 1.1

Stand: Juni 2018

## Warenzeichen

Roberta, Open Roberta und NEPO sind eingetragene Warenzeichen der Fraunhofer Gesellschaft e.V.

