



Thu Thao ist schlau



Die vierte Unterrichtseinheit führt in die Möglichkeit ein, Rechengänge mit Variablen zu kombinieren, um komplexe Skripte vereinfacht darstellen zu können.

Spätestens jetzt zeigt sich die enge Verknüpfung von Programmieren und Mathematik. Das logische Denken und Abstraktionsvermögen der Schülerinnen und Schüler wird herausgefordert und findet in einem für sie neuen Anwendungsfeld Platz.

“Was, das ist Mathe? Dann ist Mathe ab heute mein Lieblingsfach.“ Diese schöne Rückmeldung einer Schülerin der Pilotphase ist Sinnbild für die Bedeutung anwendungsbezogenen Lernens und zeigt, wie begeisterungsfähig die Schülerinnen und Schüler in dieser Altersgruppe für logisches Denken und Programmieren sind.

Überblick

Zeitaufwand	90 Minuten
Technik	touchfähige Geräte, WLAN (auch am PC mit LAN))
Methoden	Gruppenarbeit, Frage und Antwort, Simulation
Vorkenntnisse	Curriculum 1 bis 3

Kompetenzen

Die Schülerinnen und Schüler ...

- festigen ihr Wissen und ihre Umsetzungskompetenz aus den letzten Stunden, indem sie sowohl die Bruno-Schleife, als auch den Nele-Merkkasten immer wieder einsetzen und verwenden.
- lernen, dass Sie ein Computerprogramm auch nutzen können, um Rechengänge zu lösen.
- wissen am Ende, wie sie es schaffen, mithilfe von Rechengängen Strukturen zu vereinfachen.

Ablauf

Phase	Aufgabe	Methode	Zeit
Reflexion	Wiederholung der letzten Stunde	F & A	15'
Arbeitsphase	Das Winkelband	Gruppenarbeit & Plenum	70'
	Der komplizierte Mäander		
Ausblick	Die Spirale	Hausaufgabe	5'



Unterrichtsverlauf

Vorbemerkung

Die Unterrichtseinheit 4 vertieft den Einsatz von Variablen und zeigt auf, wie mit den Variablen auch gerechnet werden kann.

Achten Sie darauf, welchen Kenntnisstand im Dividieren und Multiplizieren ihre Lerngruppe hat. Die Unterrichtseinheit ist recht niedrigschwellig angelegt und arbeitet nicht mit komplizierten Ausdrücken. So wird nur mit ganzen Zahlen addiert, multipliziert und dividiert. Sie können jedoch die Aufgabenstellungen an dieser Stelle beliebig erweitern, vereinfachen oder erschweren und andere Rechengänge einbauen, wenn die Schülerinnen und Schüler bereits über ein ausreichendes Wissen verfügen. Die Herausforderungen sollten an die Lerngruppe angepasst werden.

Phase 1 | Reflexion

Stellen Sie auch zu Beginn dieser Stunde eine kleine Reflektion voran und greifen Sie das Thema der letzten Stunde wieder auf. Rufen Sie die Mäanderfabrik ins Gedächtnis und lassen Sie die Kinder noch einmal ohne Rechner die Situation simulieren. Die Kinder haben sich bestimmt den Namen Nele-Merkkasten gemerkt.

Fragen Sie die Kinder, was das besondere an der Fabrik war. Gehen Sie darauf ein, dass mithilfe des Merkkastens verschiedene Muster recht schnell in ihrer Größe variiert werden können. Dokumentieren Sie gemeinsam mit den Kindern an der Tafel, wie das Skript des Mäanders inklusive Bruno-Schleife und Nele-Merkkasten aussah. Idealerweise verwenden Sie auch hier die passenden Farben aus der Programmierumgebung.

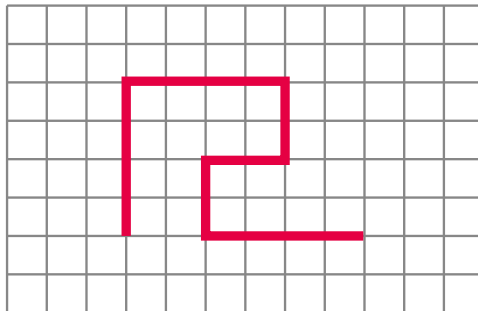
Phase 2 | Arbeitsphase

Ziel der Arbeitsphase ist es, zu erkennen, dass uns das Computerprogramm Berechnungsvorgänge abnehmen kann und so selbst komplizierte Mäander mit verschiedenen Seitenlängen mit nur einem Nele-Merkkasten dargestellt werden können.

2.1 Das Winkelband

Wir kommen an dieser Stelle auf den Mäander zurück, der in Curriculum 2 eine Rolle spielte.

Zeichnen Sie den Mäander noch einmal an die Tafel, idealerweise auf die karierten Felder. Geben Sie als Hilfestellung die Länge der Seiten an, z.B. lange Seite 200 Pixel, kurze Seite 100 Pixel.



Stellen Sie nun die Aufgabe:



Aufgabe:

Lasst die Turtle den Mäander des Tafelbildes zeichnen, so dass ein langes sogenanntes Winkelband entsteht. Nutzt dabei die Bruno-Schleife und zwei Merkkästen.

Fordern Sie nun nach und nach die Kleingruppen auf, die Größe des Mäanders zu ändern. Z.B. „Macht den Mäander mal größer in 400/200 Pixel Seitenlänge. Oder kleiner in 100/50 Seitenlänge“.

Variation: Für die schnelleren Schülerinnen und Schüler können Sie eine kleine Herausforderung einbauen: Geben Sie die Aufgabe, den Mäander mit der langen Seitenlänge 150 Pixel zu programmieren. Die kurze Seitenlänge verraten Sie vorerst nicht. Hier müssen die Kinder erst einmal überlegen. (Antwort 75 Pixel)

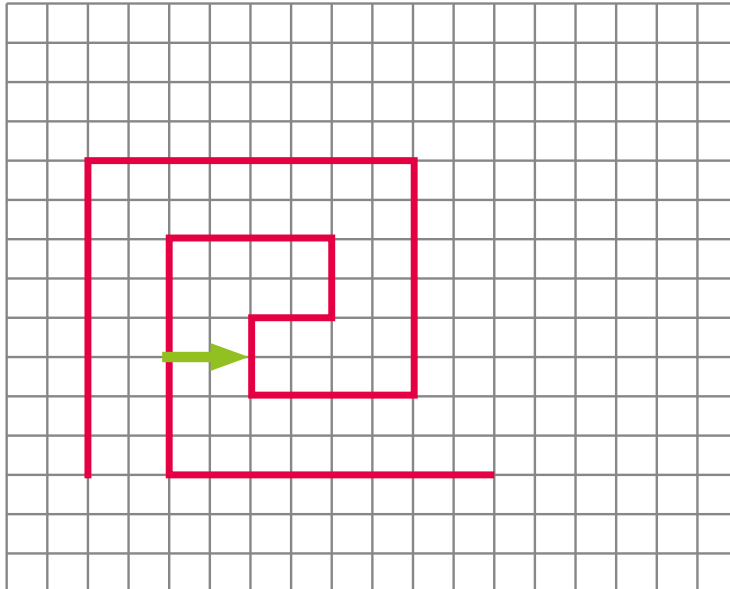
Haben alle Schülerinnen und Schüler den Mäander in mindestens zwei unterschiedlichen Größen ausgeführt, lenken Sie die Aufmerksamkeit wieder nach vorne und bereiten Sie die nächste Aufgabe vor.



2.2 Der komplizierte Mäander

Nach dieser Vorübung werden wir uns nun einem komplizierterem Muster widmen, um nach und nach zu zeigen, dass Computerprogramme rechnen können.

Zeichnen Sie folgenden Mäander an die Tafel.



Bevor die Kinder loslegen und den Mäander programmieren, überlegen Sie ein paar Fragen gemeinsam mit allen Kindern:

- Wie viele Merkkästen braucht ihr nun? Wieder nur zwei? (Antwort: Nein, 4)

Markieren Sie die Seiten und benennen Sie die jeweiligen Seitenlängen für den Nester-Merkkasten durch: laenge1, laenge2, laenge3, laenge4. Beginnen Sie mit laenge1 an der kürzesten Seite. (grüner Pfeil)

Nun können Sie die Aufgabe stellen und die Kinder erst einmal loslegen lassen.

!! Aufgabe:

Die Turtle soll das neue Muster zeichnen. Die kleinste Seite beträgt 100 Pixel.

The screenshot shows a programming environment with a script editor on the left and a drawing area on the right. The script is as follows:

```

script seltsam turtle
function main ()
  var laenge1 := 100
  var laenge2 := 200
  var laenge3 := 300
  var laenge4 := 400
  turtle → pen up
  turtle → left turn(90)
  turtle → forward(200)
  turtle → right turn(90)
  turtle → pen down
  turtle → forward(laenge4)
  turtle → right turn(90)
  turtle → forward(laenge4)
  turtle → right turn(90)
  turtle → forward(laenge3)
  turtle → right turn(90)
  turtle → forward(laenge2)
  turtle → right turn(90)
  turtle → forward(laenge1)
  turtle → right turn(90)
  turtle → forward(laenge1)
  turtle → left turn(90)

```

The drawing area on the right shows a square spiral (Mäander) with four distinct segments of increasing length, starting from the top-left corner and moving clockwise.

CyL 4.2.2 Mäander

i Hinweis! Dieser Mäander braucht etwas Zeit und die Kinder müssen buchstäblich etwas um die Ecke denken. Lassen Sie den Kindern genug Zeit und ermuntern Sie die Schnelleren, ihren Klassenkameraden zu helfen.

Sind die Ersten bereits fertig, lassen Sie die Kinder nun wieder die Größe ihres Mänders ändern.



!! Aufgabe:

Die Turtle soll das Muster noch einmal zeichnen.
Nun soll die kleinste Seite 50 Pixel betragen.

(Variante für die Schnelleren: Die kleinste Seite soll 25 Pixel betragen)

- Lassen Sie die Kinder die Aufgabe durchführen und reflektieren Sie gemeinsam mit den Kindern, an wieviel Stellen sie etwas ändern mussten. (Antwort: an 4)
- Fragen Sie die Kinder: Geht das auch mit weniger Veränderungen?
Schauen Sie sich um, bestimmt haben bei dieser Aufgabe einige Schüler einen Zettel und Stift geschnappt und haben schriftlich ausgerechnet, was an jeder Seitenlänge stehen soll.

Diese Schüler sind schlau. So wie Thu Thao, eine Schülerin aus der Pilotphase. Sie hat für die Gruppe immer auf dem Papier addiert, wie lang die Seiten sind. Sie hat also gerechnet. Thu Thao ist also schlau. Doch auch das Computerprogramm ist schlau. Es kann auch rechnen.

- Fragen Sie die Kinder: Was habt ihr gerechnet?

Überlegen Sie mit den Kindern, ausgehend von der kleinsten Seite, was jeweils gerechnet werden muss. Dokumentieren Sie dies auf der Tafel mit einer Tabelle.

	Runde 1	Runde 2	Runde 3	Rechenvorgang
laenge1	100	50	25	laenge1
laenge2	200	100	50	2*laenge1
laenge3	300	150	75	3*laenge1
laenge4	400	200	100	4*laenge1

Lassen Sie die Kinder nun die Merkkästen austauschen.

- Wie viele Merkkästen brauchen wir nun nur noch? (Antwort: Einen)

Das Skript sieht nun am Anfang folgendermaßen aus: (Der Gang der Turtle zur unteren Ecke ist in diesem Beispiel der Übersichtlichkeit halber ausgespart)

Meine Skripte Ausführen Rückgängig

```

script seltsam turtle
function main ()
  var laenge1 := 100
  turtle → forward(laenge1 *4)
  turtle → right turn(90)
  turtle → forward(laenge1 *4)
  turtle → right turn(90)
  turtle → forward(laenge1 *3)
  turtle → right turn(90)
  turtle → forward( laenge2 )
  ☹ "laenge2" kann nicht gefunden werden
  turtle → right turn(90)

```

TouchDevelop gibt Hinweise, wenn sich im Skript ein Fehler befindet. Im obigen Beispiel steckt in einem Parameter noch der alte Nele-Merkkasten. Da dieser aber nicht definiert wurde, kennt das Programm ihn nicht und gibt deshalb die Fehlermeldung aus.

Die Kinder werden in unterschiedlichem Tempo zu diesem Mäander kommen. Geben Sie ihrer Lerngruppe die Chance, das eben Gelernte zu verinnerlichen und lassen Sie die Kinder noch ein wenig mit dem Thu Thao Prinzip experimentieren und die Größen variieren. Greifen Sie gerne die Simulation der Fabrik noch einmal auf. Spannend wird es, wenn die Kinder den Mäander mit der Bruno-Schleife in lange Bänder verwandeln.

Starten Sie einen kleinen Wettbewerb:

!! Aufgabe:

Wer schafft den längsten Mäander?

Die Kinder werden versuchen, die Seitenlängen möglichst klein zu machen, um möglichst viele Wiederholungen programmieren zu können. Vielleicht schafft es sogar jemand den ganzen Bildschirm mit dem Mäander zu füllen?



 Lösungsbeispiel:

← ▶ ↺

Meine Skripte Ausführen Rückgängig

```
script wunderbar turtle (2)
function main ()
  turtle → left turn(90)
  turtle → pen up
  turtle → forward(200)
  turtle → right turn(90)
  turtle → pen down
  for 0 ≤ i < 20 do
    var laenge := 3
    turtle → forward(laenge * 4)
    turtle → right turn(90)
    turtle → forward(laenge * 4)
    turtle → right turn(90)
    turtle → forward(laenge * 3)
    turtle → right turn(90)
    turtle → forward(laenge * 2)
    turtle → right turn(90)
    turtle → forward(laenge)
    turtle → right turn(90)
    turtle → forward(laenge)
    turtle → left turn(90)
    turtle → forward(laenge)
    turtle → left turn(90)
    turtle → forward(laenge * 2)
    turtle → left turn(90)
    turtle → forward(laenge * 3)
    turtle → left turn(90)
    turtle → forward(laenge * 4)
    turtle → left turn(90)
  end for
end function
```

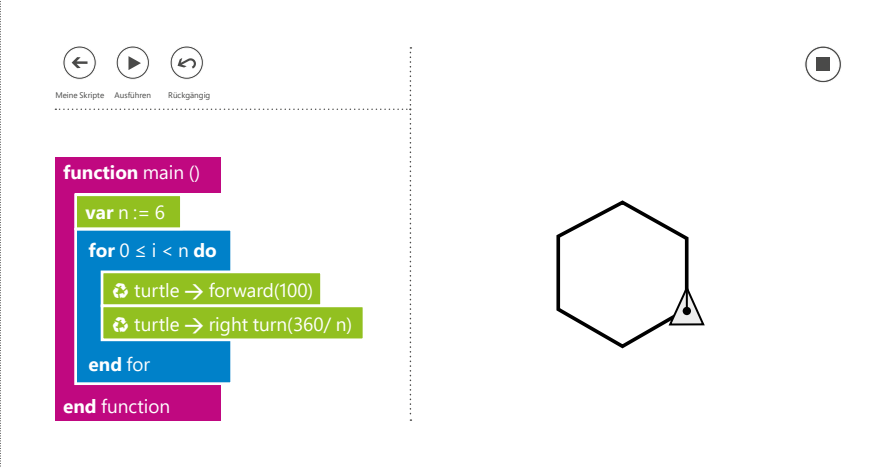


CyL 4.2.2 Mäanderband

Lassen Sie die Kinder zum Ende der Stunde selbst Mäander zeichnen und ausprobieren. Durch die Freiarbeit vertiefen die Schülerinnen und Schüler ihr Wissen und wenden das Gelernte entsprechend ihrem Können an.

Variante: Nicht nur Mäander lassen sich mit Rechengängen darstellen. Auch beliebige n-Ecke können mit schlaun Merkkästen leicht variieren. Hier verwenden wir den Rechengang Division.

Lösungsbeispiel:



The screenshot shows a turtle graphics interface. On the left, there is a code editor with the following code:

```
function main ()
  var n := 6
  for 0 ≤ i < n do
    turtle → forward(100)
    turtle → right turn(360/ n)
  end for
end function
```

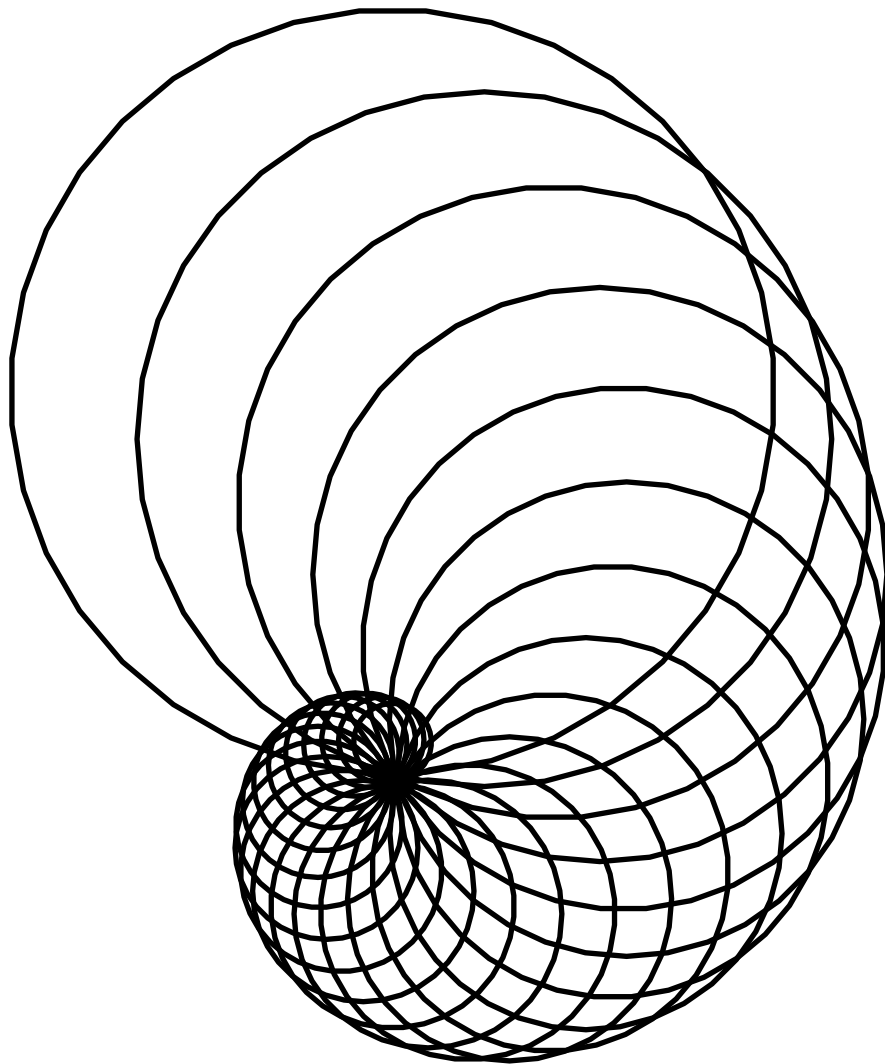
On the right, a hexagon is drawn on a white background. A small black triangle with a dot at its vertex is positioned at the bottom-right corner of the hexagon, indicating the turtle's current position and orientation.

CyL 4.2.2 6-Eck

Hinweis! Gerne können Sie den Kindern ab diesem Zeitpunkt auch zielgerichtet Knobelaufgaben geben, wie z.B. eine quadratische Schnecke oder das Haus vom Nikolaus.

2.3 Ausblick

Geben Sie den Kindern einen kleinen Ausblick auf die kommende Stunde und zeichnen Sie eine Spirale an die Tafel. Dies werden sie in der nächsten Unterrichtseinheit lernen.



Gesamtwortschatz der Turtle (Stand Tutorial 4)

🔄 turtle → pen up	Die Turtle nimmt den Stift hoch.
🔄 turtle → pen down	Die Turtle setzt den Stift ab.
🔄 turtle → left turn (90)	Die Turtle dreht sich um 90° nach links.
🔄 turtle → right turn (180)	Die Turtle dreht sich um 180° nach rechts.
🔄 turtle → forward (200)	Die Turtle läuft um 200px nach vorne.
🔄 turtle → back (100)	Die Turtle läuft um 100px rückwärts.
🔄 turtle → set pen color (colors -> red)	Die Turtle wechselt die Farbe auf „rot“.
🔄 turtle → fast	Die Turtle zeigt gleich das fertige Bild.
🔄 turtle → set speed (400)	Die Turtle läuft in vorgegebener Geschwindigkeit.
🔄 turtle → set pen color (colors->random)	Die Turtle wechselt die Farbe nach dem Zufallsprinzip.
for 0 ≤ i < 9 do	Mit der For-Schleife, auch Bruno-Schleife genannt, können Befehle beliebig oft wiederholt werden.
🔄 var laenge := 50	Im Nele-Merkkasten können Variablen als Platzhalter eingesetzt werden.