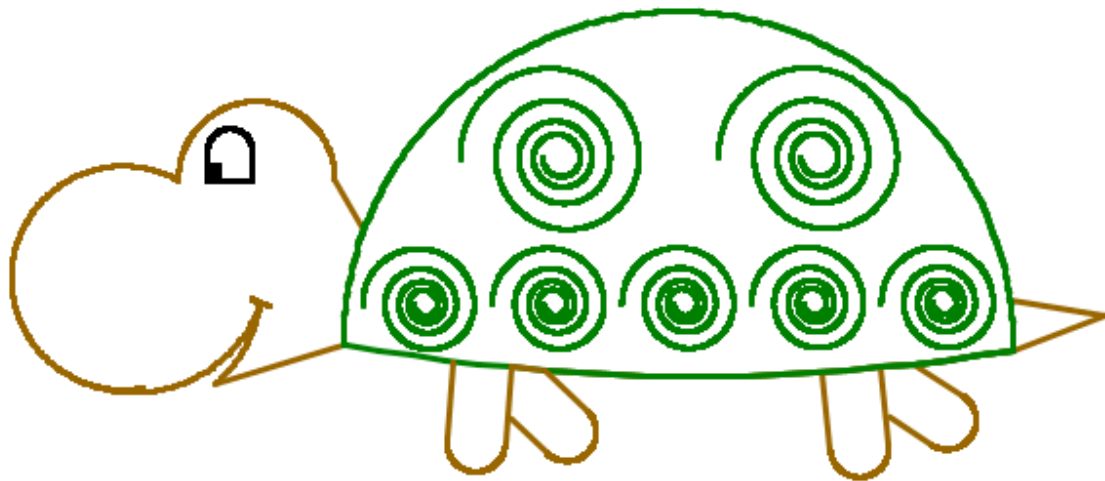


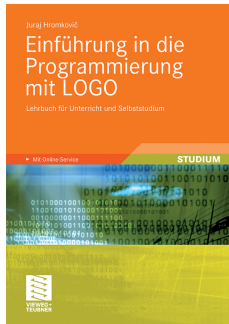
Heidi Gebauer   Juraj Hromkovič   Lucia Keller  
Giovanni Serafini   Björn Steffen

# Programmieren mit LOGO



## Programmieren mit LOGO

Dieses Skript ist eine gekürzte Version der ersten sechs Lektionen des Lehrbuches *Einführung in die Programmierung mit LOGO*. Das Lehrbuch enthält viele weitere Aufgaben und Erklärungen. Ausserdem ist es mit Hinweisen für die Lehrperson versehen. Das Lehrbuch umfasst insgesamt 14 Lektionen.



Juraj Hromkovič; *Einführung in die Programmierung mit LOGO: Lehrbuch für Unterricht und Selbststudium*, Vieweg+Teubner 2009. ISBN: 978-3-8348-1004-5.

27. Februar 2012, Version 2.0

## Programmierungsumgebung

Die vorliegenden Unterrichtsunterlagen wurden für die Programmierungsumgebung XLogo entwickelt. XLogo ist auf der Webseite [xlogo.tuxfamily.org](http://xlogo.tuxfamily.org) kostenlos verfügbar.

Damit die Logo-Programme aus den Unterlagen ausgeführt werden können, muss XLogo auf Englisch eingestellt werden.

## Nutzungsrechte

Das ABZ stellt dieses Leitprogramm zur Förderung des Unterrichts interessierten Lehrkräften oder Institutionen zur internen Nutzung kostenlos zur Verfügung.

## ABZ

Das Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich unterstützt Schulen und Lehrkräfte, die ihren Informatikunterricht entsprechend auf- oder ausbauen möchten, mit einem vielfältigen Angebot. Es reicht von individueller Beratung und Unterricht durch ETH-Professoren und das ABZ-Team direkt vor Ort in den Schulen, über Ausbildungs- und Weiterbildungskurse für Lehrkräfte, bis zu Unterrichtsmaterialien.

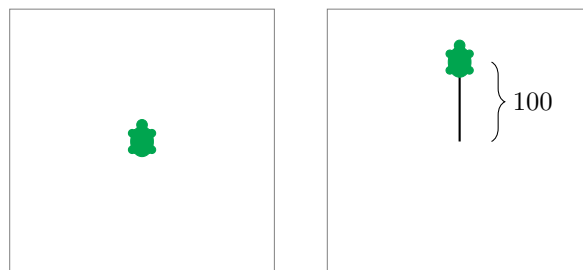
[www.abz.inf.ethz.ch](http://www.abz.inf.ethz.ch)

# 1 Grundbefehle

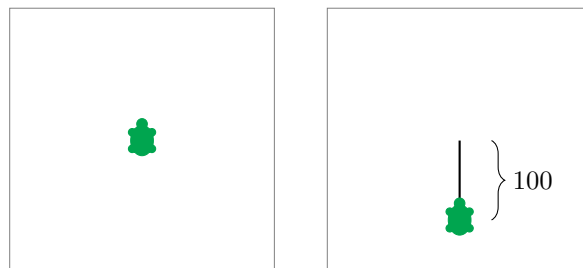
Ein **Computerbefehl** ist eine Anweisung, die der Computer versteht und ausüben kann. Der Computer kennt eigentlich nur sehr wenige Befehle und alle komplizierten Tätigkeiten, die wir vom Computer vollbracht haben wollen, müssen wir aus den einfachen Computerbefehlen zusammensetzen. Diese Folge von Computerbefehlen nennen wir **Programm**. Programme zu schreiben ist nicht immer einfach. Es gibt Programme, die aus Millionen von Befehlen zusammengesetzt sind. Hierbei die Übersicht nicht zu verlieren, erfordert ein durchdachtes und sauberes Vorgehen, das wir in diesem Programmierkurs erlernen werden.

## Gerade Linien zeichnen

Mit dem Befehl **forward 100** oder **fd 100** befehlst du der Schildkröte 100 Schritte nach vorne zu gehen:



Mit dem Befehl **back 100** oder **bk 100** geht die Schildkröte 100 Schritte zurück:



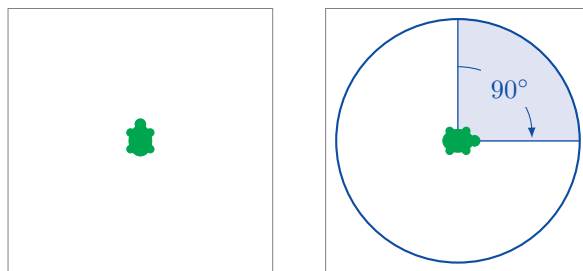
## Löschen und neu beginnen

Der Befehl **cs** löscht alles Gezeichnete, und die Schildkröte geht in die Startposition zurück.

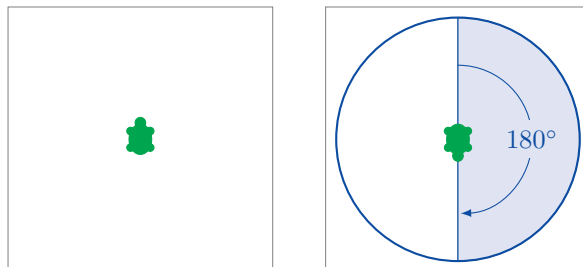
## Drehen

Die Schildkröte läuft immer in die Richtung, in die sie gerade schaut.

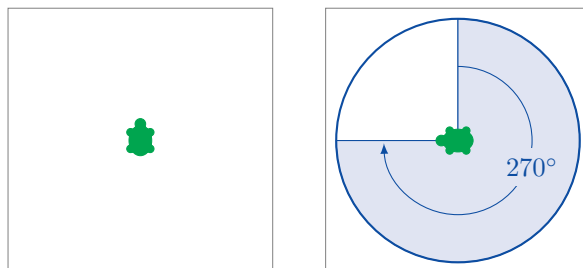
Mit dem Befehl **right 90** oder **rt 90** dreht sich die Schildkröte um  $90^\circ$  nach rechts. Dies entspricht einem Viertelkreis:



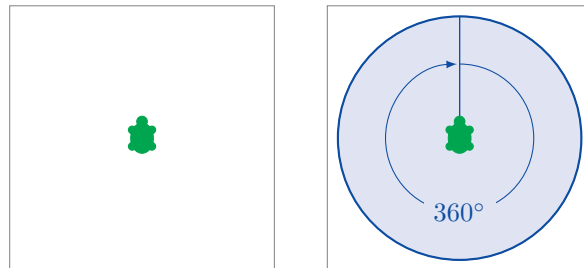
Der Befehl **right 180** oder **rt 180** dreht die Schildkröte um  $180^\circ$  nach rechts. Dies entspricht einer halben Umdrehung:



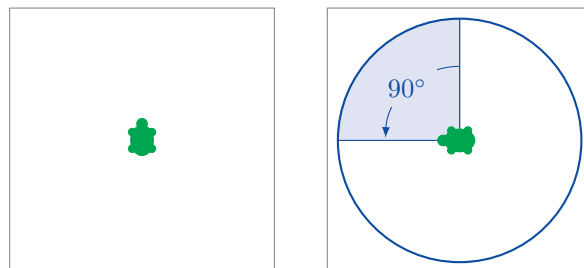
**right 270** oder **rt 270** drehen die Schildkröte um  $270^\circ$  nach rechts:



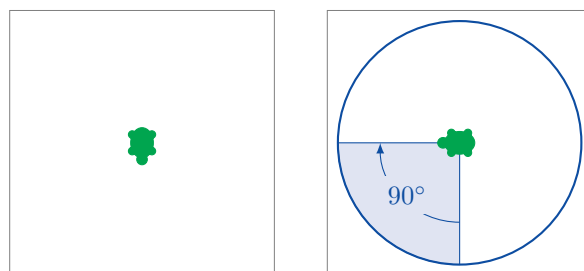
Die Befehle **right 360** und **rt 360** drehen die Schildkröte um  $360^\circ$  nach rechts. Dies entspricht einer vollen Umdrehung:



Mit dem Befehl **left 90** oder **lt 90** dreht sich die Schildkröte um  $90^\circ$  nach links:



Beachte, dass sich das Drehen nach links und rechts auf die Sicht der Schildkröte bezieht, wie das folgende Beispiel mit dem Befehl **rt 90** zeigt:



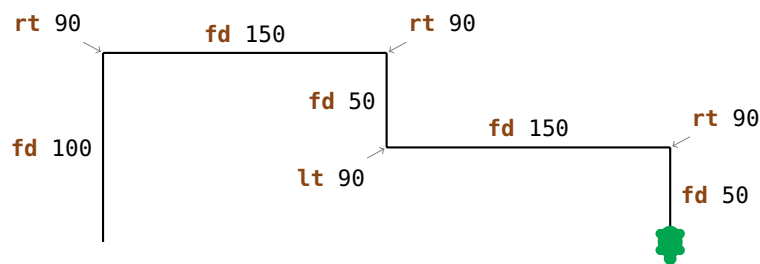
## Programmieren

**Programmieren** bedeutet, eine Folge von Computerbefehlen hintereinander zu schreiben.

 **Aufgabe 1** Tippe das folgende Programm ab und führe es aus:

```
fd 100
rt 90
fd 150
rt 90
fd 50
lt 90
fd 150
rt 90
fd 50
```


Hast du folgendes Bild gezeichnet?



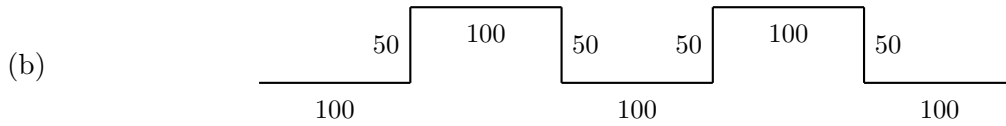
 **Aufgabe 2** Schreibe das folgende Programm und führe es aus:

```
fd 100
rt 90
fd 260
rt 90
fd 80
rt 90
fd 100
rt 90
fd 50
```

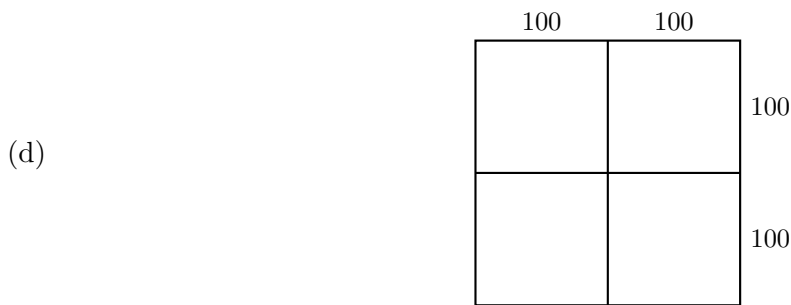
Zeichne das entstandene Bild neben das Programm oben und beschreibe (wie in Aufgabe 1) welcher Befehl, was verursacht hat.

 **Aufgabe 3** Schreibe Programme, die die folgenden Bilder zeichnen. Bei allen Bildern darfst du dir die Startposition der Schildkröte bezüglich des zu zeichnenden Bildes selbst wählen.

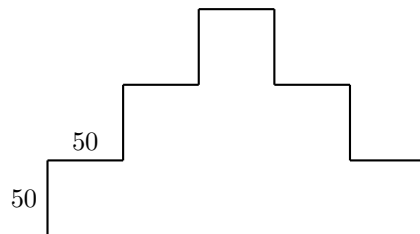




Die Größe kannst du selbst wählen.

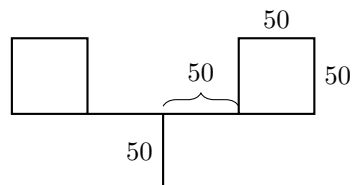


 **Aufgabe 4** Schreibe ein Programm, das das folgende Bild zeichnet:



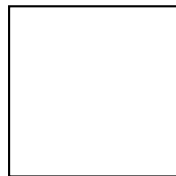
Schaffst du es, dein Programm so umzuschreiben, dass es nur die Befehle **fd 50** und **rt 90** verwendet?

 **Aufgabe 5** Anna will folgendes Bild zeichnen. Kannst du ihr helfen?



## 2 Der Befehl **repeat**

Wenn wir ein Quadrat mit Seitenlänge 100 zeichnen wollen,



dann geht das mit dem folgenden Programm:

```
fd 100  
rt 90  
fd 100  
rt 90  
fd 100  
rt 90  
fd 100  
rt 90
```

Wir beobachten, dass sich die beiden Befehle

```
fd 100  
rt 90
```

vier Mal wiederholen. Wäre es nicht einfacher, dem Computer zu sagen, dass er diese zwei Befehle vier Mal wiederholen soll anstatt diese Befehle vier Mal nacheinander aufzuschreiben?

Genau dies können wir wie folgt tun:

<b>repeat</b>	4	<b>[fd 100 rt 90]</b>
Befehl zum Wiederholen eines Programms	Anzahl der Wiederholungen	Folge von Befehlen, die wiederholt werden soll

 **Aufgabe 6** Schreibe das folgende Programm ab und führe es aus:


```
fd 75 lt 90
fd 75 lt 90
fd 75 lt 90
fd 75 lt 90
```

Was zeichnet das Programm? Kannst du den Befehl **repeat** verwenden, um das Programm kürzer zu schreiben?

 **Aufgabe 7** Tippe das folgende Programm ab, um zu sehen, was es zeichnet:

```
fd 50 rt 60
fd 50 rt 60
fd 50 rt 60
fd 50 rt 60
fd 50 rt 60
fd 50 rt 60
```

Schreibe es kürzer, indem du den Befehl **repeat** verwendest.

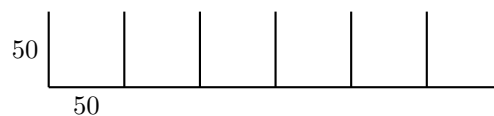
 **Aufgabe 8** Nutze den Befehl **repeat**, um ein Programm zum Zeichnen eines Quadrats mit Seitenlänge 200 zu schreiben.

 **Aufgabe 9** Schreibe das folgende Programm und führe es aus:

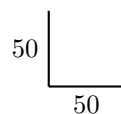
```
fd 100 rt 120
fd 100 rt 120
fd 100 rt 120
```

Was zeichnet das Programm? Kannst du den Befehl **repeat** verwenden, um das Programm kürzer zu schreiben?

Wir wollen nun das folgende Bild mit Hilfe des Befehls **repeat** zeichnen:



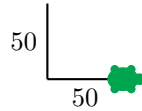
Bevor wir anfangen zu zeichnen, müssen wir uns zuerst überlegen, was das Muster ist, das sich hier wiederholt. Zum Beispiel können wir als das zu wiederholende Muster das folgende Bild nehmen:



Wenn wir in der unteren linken Ecke starten, können wir das Muster mit folgendem Programm zeichnen:

```
fd 50 bk 50 rt 90 fd 50
```

Nach der Ausführung dieses Programms steht die Schildkröte wie in der folgenden Abbildung gezeichnet und schaut nach rechts:

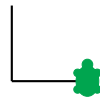


Das ist gut, weil sie schon die Position erreicht hat, an der wir das Muster nochmals zeichnen können. Sie braucht nur noch nach oben zu schauen, was wir mit dem Befehl **lt 90** erreichen können.

Wir führen das Programm aus, um zu überprüfen, ob es stimmt:

```
fd 50 bk 50 rt 90 fd 50  
lt 50
```

Wir erhalten die gewünschte Situation:



Wenn wir das gleiche Programm nochmals laufen lassen, erhalten wir:



Wir sehen also, dass unsere Idee funktioniert und können unser Programm 5 Mal wiederholen:

```
repeat 5 [ fd 50 bk 50 rt 90 fd 50 lt 90 ]
```

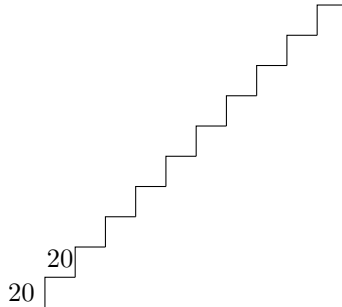
Muster                      Ausrichten

Viele Aufgaben können wir mit dieser Vorgehensweise lösen. Alle folgenden Aufgaben funktionieren so. Denke immer daran, dass du zuerst das *Muster*, welches sich wiederholt, finden musst. Danach musst du einerseits ein Programm für das *Muster* und andererseits ein Programm für das *Ausrichten* der Schildkröte zum Zeichnen des nächsten *Musters* entwickeln. Dein Programm sieht dann wie folgt aus.

```
repeat Anzahl [ Muster Ausrichten ]
```

 **Aufgabe 10** Zeichnen von Treppen.


(a) Zeichne eine Treppe mit 10 Stufen der Grösse 20:



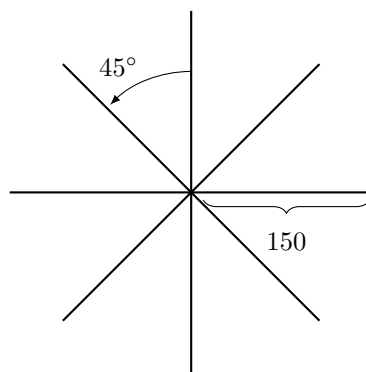
- Finde zuerst das Muster, welches sich wiederholt und schreibe ein Programm dafür.
- Überlege dir ein Programm für das Ausrichten der Schildkröte, damit sie für die nächste Wiederholung des Musters richtig steht.
- Setze dann die beiden Programme dann geeignet zusammen, um die Aufgabe damit zu lösen.

(b) Zeichne eine Treppe mit 5 Stufen der Grösse 50.

(c) Zeichne eine Treppe mit 20 Stufen der Grösse 10.

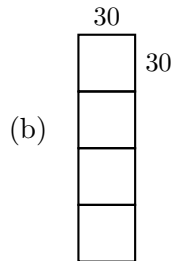
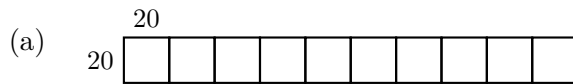
 **Aufgabe 11** Nun wollen wir Sterne zeichnen.


(a) Zeichne den folgenden Stern:

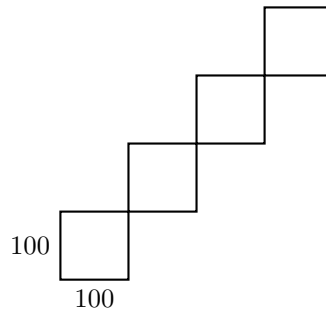


(b) Der Stern hat acht Strahlen der Länge 150. Kannst du auch einen Stern mit 16 Strahlen der Länge 100 zeichnen?

 **Aufgabe 12** Zeichne die folgenden Bilder:



 **Aufgabe 13** Zeichne mit einem Programm das folgende Bild:



## Wandermodus

Normalerweise befindet sich die Schildkröte im **Stiftmodus**. Das heisst, sie hat einen Stift in der Hand und zeichnet immer, wenn sie sich bewegt.

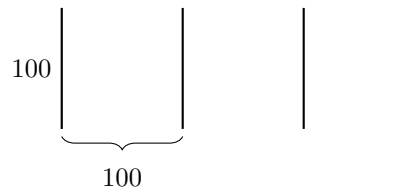
Im **Wandermodus** bewegt sich die Schildkröte auf dem Bildschirm ohne zu zeichnen. In den Wandermodus kommst du mit dem Befehl

**penup** oder kurz **pu**.

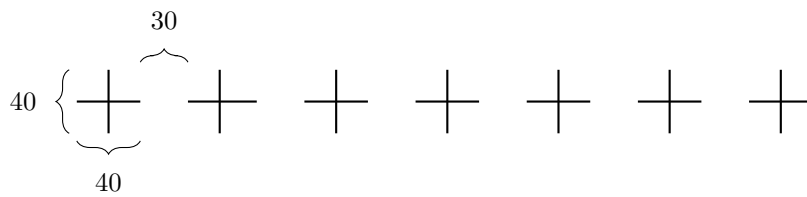
Aus dem Wandermodus zurück in den Stiftmodus kommst du mit dem Befehl

**pendown** oder kurz **pd**.

 **Aufgabe 14** Zeichne das folgende Bild mit einem Programm:



 **Aufgabe 15** Schreibe ein Programm für das folgende Bild:



 **Aufgabe 16** Schreibe das folgende Programm ab und führe es aus:

```
repeat 4 [fd 100 rt 90]
rt 90
repeat 4 [fd 100 rt 90]
rt 90
repeat 4 [fd 100 rt 90]
rt 90
repeat 4 [fd 100 rt 90]
rt 90
```

Was entsteht dabei? Schaffst du es, dieses Programm noch kürzer zu schreiben?

## 3 Programme benennen und aufrufen

Jedem Programm, das wir geschrieben haben, können wir einen Namen geben. Wenn wir dann den Namen des Programms in die Befehlszeile schreiben, wird die Tätigkeit des Programms durchgeführt.

Das Programm zum Zeichnen eines Quadrats mit Seitenlänge 100 ist:

```
repeat 4 [fd 100 rt 90]
```

Wir können dieses Programm auf folgende Weise mit dem Namen **QUADRAT100** versehen:

```
to QUADRAT100
repeat 4 [fd 100 rt 90]
end
```

Wir haben also zweimal dasselbe Programm geschrieben, einmal ohne und einmal mit Namen.

Programme mit Namen schreiben wir in den **Editor**. Solche Programme sind in diesem Heft durch eine graue Box markiert. Sobald unser Programm fertig ist, drücken wir auf den Knopf mit der Schildkröte, um den Editor wieder zu schliessen.

Den Namen kann sich jeder selbst aussuchen, wir haben **QUADRAT100** gewählt, weil wir andeuten wollen, dass es um das Zeichnen eines Quadrats mit Seitenlänge 100 geht. Die einzigen Bedingungen für die Namen sind, dass sie mit einem Buchstaben beginnen und in einem Stück, ohne Abstand dazwischen, geschrieben werden.

Auf dem Bildschirm wurde noch nichts gezeichnet, weil wir dem Programm nur einen Namen gegeben haben, es aber noch nicht ausgeführt haben. Wenn wir nun den Namen

**QUADRAT100**

in die Befehlszeile schreiben, dann wird das Programm **repeat 4 [fd 100 rt 90]** ausgeführt. Auf dem Bildschirm erscheint:



Schauen wir uns nochmals die Aufgabe 12 (a) an. Wir können diese Aufgabe einfacher lösen, indem wir zuerst ein Programm schreiben für das zu wiederholende Muster, also das Quadrat mit Seitenlänge 20, und ihm einen Namen geben:

```
to QUADRAT20
repeat 4 [fd 20 rt 90]
end
```

Nach dem Zeichnen von `QUADRAT20` steht die Schildkröte in der linken unteren Ecke des Quadrats:



Um das nächste Quadrat zu zeichnen, muss sie in die rechte untere Ecke kommen. Dies erreichen wir durch das Programm


```
rt 90 fd 20 lt 90
```

Wir werden dieses Programm ebenfalls benennen:

```
to AUSRICHTEN20
rt 90 fd 20 lt 90
end
```

Mit diesen zwei Programmen können wir ein Programm für die Aufgabe 12 (a) wie folgt schreiben:

```
repeat 10 [QUADRAT20 AUSRICHTEN20]
```

 **Aufgabe 17** Schreibe ein Programm zur Lösung der Aufgabe 12 (b), das ein Programm zum Zeichnen der Quadrate mit Seitenlängen 30 verwendet. Das Programm muss wie folgt aussehen:

```
repeat 4 [QUADRAT30 AUSRICHTEN30]
```

Du musst also die passenden Programme `QUADRAT30` und `AUSRICHTEN30` schreiben.

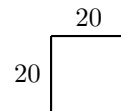
Unser vorheriges Programm können wir auch benennen. Zum Beispiel:

```
to REIHE10
repeat 10 [QUADRAT20 AUSRICHTEN20]
end
```

Wenn wir dies so tun, nennen wir die Programme `QUADRAT20` und `AUSRICHTEN20` **Unterprogramme** des Programms `REIHE10`.

 **Aufgabe 18** Nutze das Programm `QUADRAT100` als Unterprogramm zum Zeichnen des Bildes aus Aufgabe 13.

 **Aufgabe 19** Schreibe ein Programm zum Zeichnen einer Treppenstufe



und nutze es als Unterprogramm zur Lösung der Aufgabe 10 (a).

 **Aufgabe 20**

(a) Löse Aufgabe 11 (a) nochmals indem du folgendes Unterprogramm verwendest:

```
to LINIE
fd 150 bk 150
end
```

(b) Löse Aufgabe 11 (b) nochmals indem du das folgende Unterprogramm verwendest:

```
to STRAHL
fd 150 bk 300 fd 150
end
```

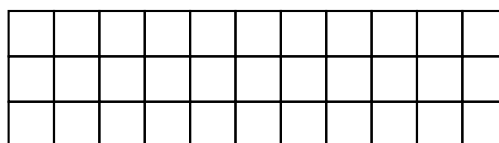
 **Aufgabe 21** Löse die Aufgabe 14 und die Aufgabe 15 nochmals mit Hilfe von Unterprogrammen.

 **Aufgabe 22** Wir haben vorher ein Programm `REIHE10` erstellt. Was macht das folgende Programm?

```
REIHE10 fd 20 lt 90 fd 200 rt 90
```

Überprüfe deine Idee auf dem Computer.

 **Aufgabe 23** Schreibe ein Programm, welches das folgende Bild zeichnet:

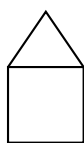



# Häuser bauen

Als nächstes möchten wir einem Architekten beim Bau einer Wohnsiedlung helfen. Damit der Bau einfacher ist, möchte er alle Häuser gleich bauen. Wir geben ihm den folgenden Vorschlag:

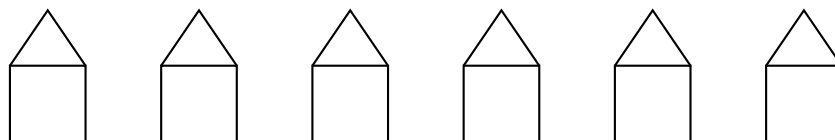
```
to HAUS
rt 90
repeat 4 [fd 50 rt 90]
lt 60 fd 50 rt 120 fd 50 lt 150
end
```

Dieses Programm zeichnet das folgende Haus:



 **Aufgabe 24** Wo startet die Schildkröte mit dem Zeichnen des Hauses? Überlege dir den Weg, den die Schildkröte zurücklegt, beim Zeichnen des obigen Hauses mit Hilfe des Programms **HAUS**. Wo steht die Schildkröte, nachdem sie das Haus gezeichnet hat? Zeichne das entsprechende Bild und beschreibe wie in Aufgabe 1 welcher Befehl, welche Wirkung hatte.

Der Architekt hat dieses Haus aufbauen lassen und sieht jetzt, dass alles funktioniert. Deshalb verwendet er jetzt dieses Programm als Baustein, um eine erste Strasse mit Häusern zu bebauen. Am Schluss soll die Strasse so aussehen:



Weil er das Haus immer nach dem gleichen Muster zeichnet, kann er den Baustein **HAUS** 6 Mal verwenden und muss sich nicht jedes Mal neu überlegen, wie er das Haus bauen soll. Er lässt die Schildkröte zuerst das erste Haus von links zeichnen und sagt der Schildkröte danach, dass sie an den Anfangspunkt des zweiten Hauses springen soll:



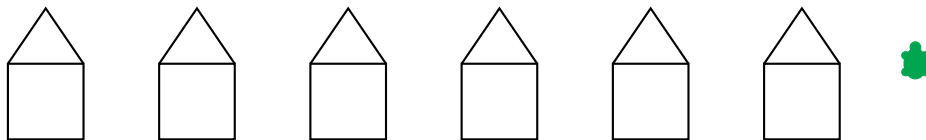
Das macht der Architekt mit dem folgenden Programmteil:


```
HAUS rt 90 pu fd 50 lt 90 pd
```

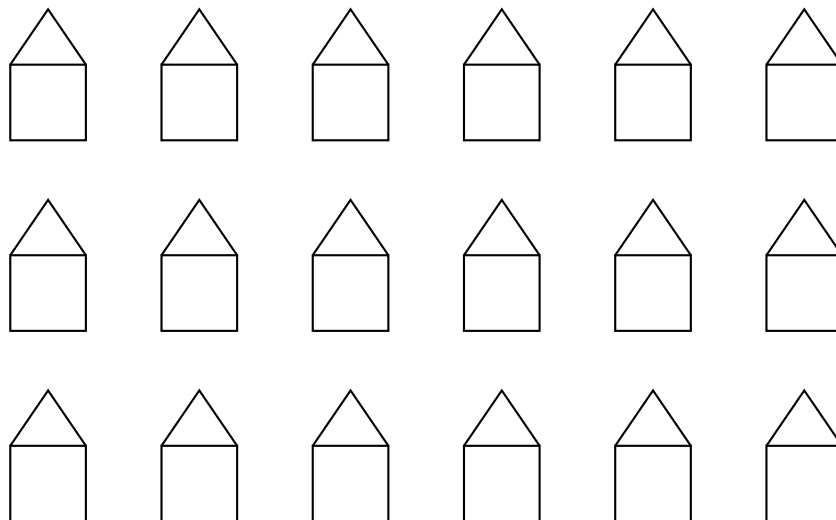
Nun könnte die Schildkröte an diesem Ort genau das gleiche Haus nochmals zeichnen und wieder an den Anfangspunkt des nächsten Hauses springen. Das macht sie so lange, bis sie alle 6 Häuser gezeichnet hat. Wir müssen also den Programmteil oben 6 Mal wiederholen und bekommen dann eine Reihe mit 6 gleichen Häusern. Das Programm dazu nennen wir **HAUSREIHE**:

```
to HAUSREIHE  
repeat 6 [HAUS rt 90 pu fd 50 lt 90 pd]  
end
```

Die Schildkröte steht ganz am Schluss da, wo das nächste Haus gezeichnet werden würde:



 **Aufgabe 25** Nun möchten wir die Wohnsiedlung noch um ein paar Strassen erweitern. Benutze das Programm **HAUSREIHE** als Baustein, um das folgende Bild zu zeichnen:



*Hinweis:* Nach jeder Reihe muss die Schildkröte an die richtige Stelle springen, um die nächste Reihe zu zeichnen.

# Dicke Linien und schwarze Quadrate

 **Aufgabe 26** Dicke Linien zeichnen mit dem Programm **FETT**

(a) Benenne das folgende Programm **FETT**

```
fd 100
rt 90
fd 1
rt 90
fd 100
rt 180
```

und schreibe dann

**FETT**

Was zeichnet die Schildkröte? Zeichne mit dem Bleistift auf ein Blatt, wie das Bild entstanden ist.

(b) Wiederhole 99 Mal das Programm **FETT** mit dem Programm

```
repeat 99 [FETT]
```

Was entsteht auf dem Bildschirm? Wieso wiederholen wir nur 99 Mal und nicht 100 Mal?

 **Aufgabe 27** Unterschiedlich grosse Quadrate zeichnen.

(a) Schreibe ein Programm, das ein Quadrat mit Seitenlänge 50 zeichnet und benenne es **QUADRAT50**. Probiere es aus, um zu sehen, ob es das Richtige macht.


(b) Schreibe ein Programm, das ein Quadrat mit Seitenlänge 75 zeichnet.

(c) Lasse das Programm

```
QUADRAT50
QUADRAT75
QUADRAT100
```

ausführen. Was entsteht dabei?

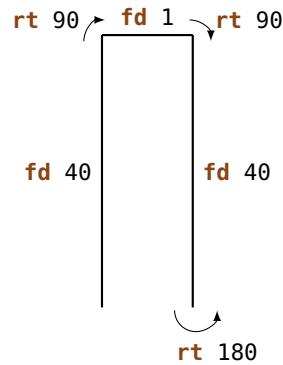
(d) Wie würdest du das Programm ändern, um noch drei weitere, grössere Quadrate hinzu zu zeichnen?

 **Aufgabe 28** In dieser Aufgabe werden wir fette Linien und schwarze Flächen zeichnen. In Aufgabe 26 haben wir gesehen, dass eine fette Linie wie folgt gezeichnet werden kann:

```

to FETT40
fd 40
rt 90
fd 1
rt 90
fd 40
rt 180
end

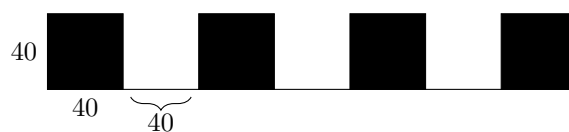
```



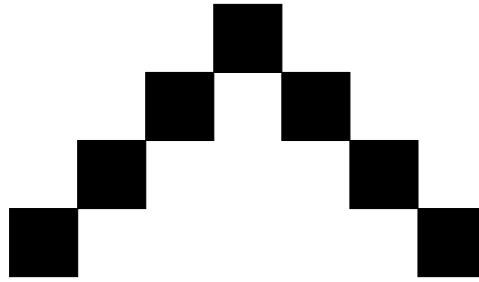
Die fette Linie entsteht, indem man zwei Linien so dicht nebeneinander zeichnet, dass die beiden Linien zusammen wie eine dicke Linie aussehen.

- (a) Tippe das Programm `FETT40` ab und probiere es aus.
- (b) Eine fette Linie der Länge 40 kann man als ein Rechteck mit Breite 2 und Länge 40 betrachten. Nach dem Zeichnen von `FETT40` steht die Schildkröte bei der zweiten Linie unten und schaut nach oben. Wenn das Programm `FETT40` also wiederholt wird, übermalt die Schildkröte diese zweite Linie. Wir bekommen dann also ein Rechteck mit Breite 3 und Länge 40. Mit jeder Wiederholung kommt also nur eine neue Linie hinzu. Wenn wir `FETT40` 39 Mal wiederholen, entsteht das schwarze Quadrat mit Seitenlänge 40. Probiere es aus, indem du `FETT40` 39 Mal wiederholst.
- (c) Schreibe ein Programm mit dem Namen `SCHWARZ40`, das ein schwarzes Quadrat mit Seitenlänge 40 zeichnet.

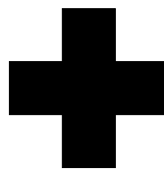
 **Aufgabe 29** Nutze das Programm `SCHWARZ40`, um das folgende Bild zu zeichnen:



 **Aufgabe 30** Verwende das Programm `SCHWARZ40`, um das folgende Bild zu zeichnen:





 **Aufgabe 31** Zeichne das folgende Bild:



 **Aufgabe 32** Schreibe ein Programm, um das folgende Bild zu zeichnen:



 **Aufgabe 33** Der Architekt beschliesst, das Dach bei einem anderen Lieferanten zu bestellen. Er bekommt also zwei Bausteine: Einen Baustein `DACH` und einen Baustein `UNTERERTEIL`. Schreibe für den Architekten zwei Programme, die diese zwei Bausteine zeichnen, und setze sie dann in einem neuen Programm `HAUS1` zu einem Haus zusammen.

 **Aufgabe 34** Die Häuser in Aufgabe 25 sind ziemlich einfach gebaut. Sei kreativ und entwerfe ein neues Haus und baue damit eine neue Wohnsiedlung.

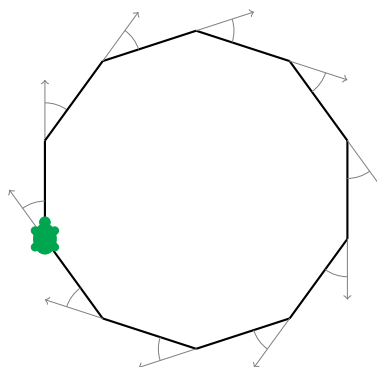
# 4 Regelmässige Vielecke und Kreise

## Regelmässige Vielecke

Ein regelmässiges  $k$ -Eck hat  $k$  Ecken und  $k$  gleich lange Seiten. Wenn du ein Vieleck, zum Beispiel ein 10-Eck, mit Bleistift zeichnen möchtest, musst du 10 Linien zeichnen und nach jeder Linie „ein bisschen“ die Richtung ändern (drehen).

Wie viel muss man drehen?

Wenn man ein Vieleck zeichnet, dreht man mehrmals unterwegs, aber am Ende steht man genau an der gleichen Stelle und schaut in genau die gleiche Richtung wie am Anfang.



Das bedeutet, dass man sich unterwegs volle  $360^\circ$  gedreht hat. Wenn man also ein regelmässiges 10-Eck zeichnet, hat man sich genau zehn Mal gedreht, und zwar immer um einen gleich grossen Winkel. Dieser Winkel ist:

$$\frac{360^\circ}{10} = 36^\circ$$

Daher muss man sich immer um  $36^\circ$  drehen: **rt 36**. Probieren wir das, indem wir folgendes Programm schreiben:

```
repeat 10 [ fd 50      rt 36 ]  
           Seitenlänge  Drehung um 36°
```

 **Aufgabe 35** Zeichne die folgenden regelmässigen Vielecke:

- (a) ein 5-Eck mit der Seitenlänge 180,
- (b) ein 12-Eck mit der Seitenlänge 50,
- (c) ein 4-Eck mit der Seitenlänge 200,
- (d) ein 6-Eck mit der Seitenlänge 100,
- (e) ein 3-Eck mit der Seitenlänge 200 und
- (f) ein 18-Eck mit der Seitenlänge 20.

Wenn man ein 7-Eck zeichnen will, hat man das Problem, dass man 360 nicht ohne Rest durch 7 teilen kann. In diesem Fall lässt man das Resultat durch den Computer ausrechnen, indem man

`360/7`

schreibt („/“ bedeutet für den Computer „teile“). Der Computer findet dann das genaue Resultat. Somit kann man ein 7-Eck mit Seitenlänge 100 wie folgt zeichnen:

`repeat 7 [fd 100 rt 360/7]`

Probiere es aus.

## Kreise zeichnen

Mit den Befehlen `fd` und `rt` kann man keine genauen Kreise zeichnen. Wie du aber sicherlich beobachtet hast, sehen Vielecke mit vielen Ecken Kreisen sehr ähnlich. Wenn wir also viele Ecken und sehr kurze Seiten nehmen, erhalten wir dadurch Kreise.


 **Aufgabe 36**

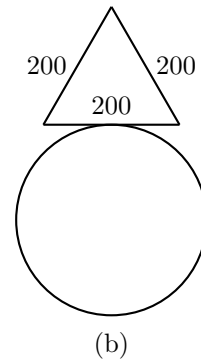
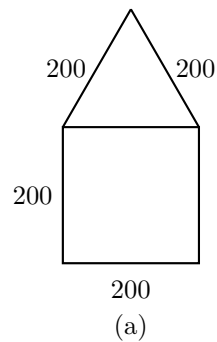
- (a) Teste die folgenden Programme:

```
repeat 360 [fd 1 rt 1]
repeat 180 [fd 3 rt 2]
repeat 360 [fd 2 rt 1]
repeat 360 [fd 3.5 rt 1]
```

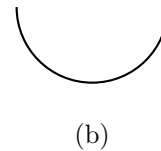
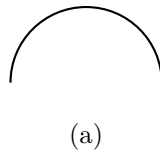
3.5 bedeutet 3 und einen halben Schritt.

- (b) Was würdest du tun, um einen ganz kleinen Kreis zu zeichnen? Schreibe ein Programm dafür.
- (c) Was würdest du tun, um einen grossen Kreis zu zeichnen? Schreibe ein Programm dafür.

-  **Aufgabe 37** Nutze deine neuen Erfahrungen, um die folgenden Bilder zu zeichnen. Die Grösse des Kreises darfst du selber wählen:



-  **Aufgabe 38** Versuche, die folgenden Halbkreise zu zeichnen. Die Grössen darfst du selber bestimmen:



## Fantasiemuster

Zeichne ein 7-Eck mit:

```
repeat 7 [fd 100 rt 360/7]
```

Drehe dann die Schildkröte um  $10^\circ$  mit


```
rt 10
```

Wiederhole beides ein paar Mal, und schau dir das Bild an. Wir drehen nach jedem 7-Eck immer um  $10^\circ$  mit `rt 10`. Wenn wir wieder in die Ausgangsposition zurückkommen wollen, dann müssen wir diese Tätigkeit

$$\frac{360^\circ}{10^\circ} = 36$$

Mal wiederholen. Also schauen wir uns an, was das folgende Programm zeichnet:


```
repeat 36 [repeat 7 [fd 100 rt 360/7] rt 10]
```

 **Aufgabe 39** Zeichne ein regelmässiges 12-Eck mit Seiten der Länge 70 und drehe es 18 Mal, bis du wieder an die Startposition kommst.

*Hinweis:* Du kannst zuerst ein Programm für ein 12-Eck mit Seitenlänge 70 schreiben und ihm zum Beispiel den Namen `ECK12` geben. Dann musst du nur noch das Programm


















```
repeat 18 [ECK12 rt ... ]
```

vervollständigen.

 **Aufgabe 40** Denke dir eine ähnliche Aufgabe wie Aufgabe 39 aus, und schreibe ein Programm dazu.

## Farben

Wenn man schon Fantasiemuster zeichnet, passen dazu auch Farben. Die Schildkröte kann nicht nur mit Schwarz, sondern mit einer beliebigen Farbe zeichnen. Jede Farbe ist durch eine Zahl bezeichnet. Eine Übersicht aller Farben findest du in der folgenden Tabelle:

Farbnummer	Farbe	Farbnummer	Farbe
0		9	
1		10	
2		11	
3		12	
4		13	
5		14	
6		15	
7		16	
8			

Mit dem Befehl

`setpencolor`

`X`

Befehl zum  
Ändern  
der Farbe

Nummer der ge-  
wünschten Farbe

wechselt die Schildkröte von der aktuellen Farbe in die Farbe mit der Nummer `X`. Wir können den Befehl durch `setpc` abkürzen.

Damit kann man tolle Muster zeichnen, wie zum Beispiel das Muster, das durch das folgende Programm entsteht. Zuerst benennen wir zwei Programme zum Zeichnen zweier Kreise mit unterschiedlicher Grösse:

```
to KREIS3
repeat 360 [fd 3 rt 1]
end

to KREIS1
repeat 360 [fd 1 rt 1]
end
```

Jetzt nutzen wir diese Kreise, um ähnliche Muster wie die bisherigen zu entwerfen:

```
to MUST3
repeat 36 [KREIS3 rt 10]
end

to MUST1
repeat 18 [KREIS1 rt 20]
end
```

Jetzt versuchen wir es mit Farben:


```
setpc 2
MUST3 rt 2
setpc 3
MUST3 rt 2

setpc 4
MUST3 rt 2
setpc 5
MUST3 rt 2

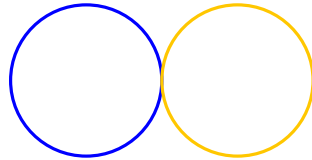
setpc 6
MUST1 rt 2
setpc 15
MUST1 rt 2

setpc 8
MUST1 rt 2
setpc 9
MUST1 rt 2
```

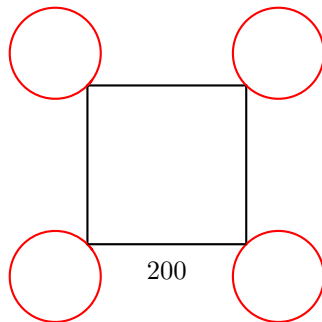
Du darfst gerne die Arbeit fortsetzen und noch mehr dazu zeichnen. Oder zeichne ein Muster nach eigener Vorstellung.

 **Aufgabe 41** Nutze `MUST3`, um das entsprechende Bild mit der orangenen Farbe zu zeichnen. Verwende danach den Befehl `setpc 7`, um zur weissen Farbe zu wechseln. Was passiert jetzt, wenn du wieder `MUST3` ausführen lässt?

 **Aufgabe 42** Zeichne das folgende Bild. Die Schildkröte ist am Anfang am gemeinsamen Punkt (im Schnittpunkt) der beiden Kreise.




 **Aufgabe 43** Schreibe ein Programm zum Zeichnen des folgenden Bildes. Die Kreisgrösse darfst du selber wählen.





Wir nennen **:ECK** einen **Parameter**. Im Beispiel oben sind 3, 4, 5 und 6 **Werte des Parameters :ECK**. Der Computer erkennt den Parameter am **:**. Deshalb muss überall, wo ein Parameter vorkommt, ein **:** vor dem Namen des Parameters stehen.

 **Aufgabe 44** Die folgenden Programme zeichnen Quadrate verschiedener Seitenlängen:


```
repeat 4 [fd 100 rt 90]
repeat 4 [fd 50 rt 90]
repeat 4 [fd 200 rt 90]
```

Die gelben Zahlen 100, 50, 200 kann man als Werte eines Parameters betrachten, der die Grösse des Vierecks bestimmt. Schreibe ein Programm mit dem Parameter **:GR**, das ein beliebig grosses Viereck zeichnen kann.


 **Aufgabe 45** Die folgenden Programme zeichnen unterschiedlich grosse Kreise:

```
repeat 360 [fd 1 rt 1]
repeat 360 [fd 12 rt 1]
repeat 360 [fd 3 rt 1]
```

Schreibe ein Programm mit einem Parameter, mit dem man beliebig grosse Kreise zeichnen kann, und probiere es für die Parametergrössen 1, 2, 3, 4 und 5 aus. Den Namen des Programms und den Namen des Parameters darfst du dir selber aussuchen. Du musst nur aufpassen, dass immer der Doppelpunkt vor dem Parameter steht.


 **Aufgabe 46** Erinnerst du dich noch daran, wie man fette Linien zeichnen kann (Aufgabe 26)? Schreibe ein Programm mit einem Parameter, das eine fette Linie beliebiger Länge zeichnen kann.

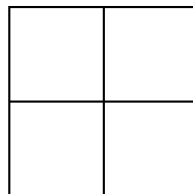
*Hinweis:* Du kannst zuerst ein Programm für eine Linie der Länge 100 und ein Programm für eine Linie der Länge 50 schreiben, um zu erkennen, wo der Parameter liegt.



 **Aufgabe 47** Schreibe ein Programm mit einem Parameter, das ein beliebig grosses Dreieck zeichnet. Zeichne dann mit diesem Programm nacheinander Dreiecke der Grössen

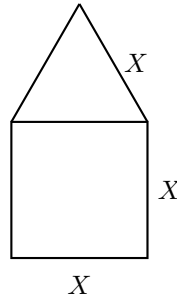
20, 40, 60, 80, 100, 120, 140, 160 und 180.

Was entsteht dabei?

 **Aufgabe 48** Schreibe ein Programm, welches das folgende Bild mit frei wählbarer Grösse zeichnet:



-  **Aufgabe 49** Schreibe ein Programm mit einem Parameter, das Sechsecke beliebiger Seitenlängen zeichnet. Probiere das Programm zum Zeichnen von Sechsecken für die Seitenlängen 40, 60 und 80 aus.
-  **Aufgabe 50** Schreibe ein Programm mit einem Parameter `:X`, das beliebig grosse Häuser wie in der folgenden Abbildung zeichnet.



## Programm mit mehreren Parametern

Ein Programm kann mehr als einen Parameter haben. Wenn wir Vielecke zeichnen, können wir einen Parameter `:ECK` für die Anzahl der Ecken und einen Parameter `:GR` für die Seitenlänge bestimmen.

In den folgenden Programmen ist der Parameter `:ECK` mit gelb und der Parameter `:GR` mit grün markiert:


```
repeat 13 [fd 100 rt 360/13]
repeat 3 [fd 300 rt 360/3]
repeat 17 [fd 10 rt 360/17]
repeat 60 [fd 3 rt 360/60]
```

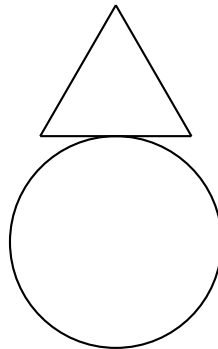
Damit können wir jetzt ein Programm für unterschiedliche Vielecke schreiben:


```
to VECKE :ECK :GR
repeat :ECK [fd :GR rt 360/:ECK]
end
```

Teste das Programm `VECKE` mit den folgenden Aufrufen:

```
VECKE 12 60
VECKE 12 45
VECKE 8 30
VECKE 9 30
VECKE 7 31
VECKE 11 50
```

-  **Aufgabe 51** Schreibe ein Programm mit zwei Parametern, welches das folgende Bild zeichnen kann. Dabei soll die Kreisgrösse sowie die Grösse des Dreiecks frei wählbar sein.



-  **Aufgabe 52** Das Programm

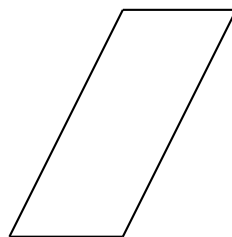
```
fd 100 rt 90 fd 200 rt 90 fd 100 rt 90 fd 200
```

zeichnet ein Rechteck mit Breite 100 und Länge 200. Überprüfe es und schreibe ein Programm mit zwei Parametern, so dass Rechtecke mit beliebig grossen Breiten und Längen gezeichnet werden können.

-  **Aufgabe 53** Das folgende Programm

```
rt 45 fd 200 rt 45 fd 100  
rt 90 rt 45 fd 200 rt 45 fd 100
```

zeichnet ein Parallelogramm:



Schreibe ein Programm mit zwei Parametern, welches solche Parallelogramme mit beliebigen Seitenlängen zeichnen kann.

 **Aufgabe 54** Zeichne eine Blume, indem du einen Kreis mit

`VECKE 360 2`

zeichnest, die Schildkröte dann ein bisschen drehst mit

`rt 20`

und anschliessend wieder einen Kreis zeichnest mit

`VECKE 360 2`

und so weiter fortfährst mit `rt 20 VECKE 360 2 rt 20 VECKE 360 2 ...`

Wenn du die Blume fertig gezeichnet hast, steht die Schildkröte wieder auf der ursprünglichen Position. Die Schildkröte hat also 18 Kreise gezeichnet und sich dazwischen jeweils um  $20^\circ$  gedreht, somit hat sich die Schildkröte insgesamt um  $18 \times 20^\circ = 360^\circ$  gedreht.

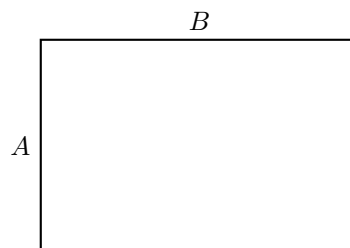
Zusammengefasst ergibt dies das folgende Programm:

`repeat 18 [VECKE 360 2 rt 20]`

Probiere es aus.

- Du kannst aber auch Blumen mit 10 Blättern (Kreisen) oder mit 20 Blättern (Kreisen) zeichnen. Wie würdest du das machen? Schreibe ein Programm dazu und probiere es aus.
- Kannst du ein Programm mit einem Parameter schreiben, mit dem man Blumen mit beliebig vielen Blättern (Kreisen) zeichnen könnte?
- Schaffst du es, ein Programm zu schreiben, mit dem du folgende Parameter frei wählen kannst:
  - die Anzahl der Blätter (Kreise) und
  - die Grösse der Kreise?

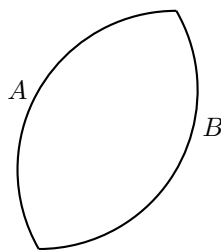
 **Aufgabe 55** Schreibe ein Programm zum Zeichnen beliebiger Rechtecke in beliebiger Farbe:



Dies bedeutet, dass die Seitenlängen *A* und *B* sowie die Farbe frei wählbar sind.

## 6 Blumen zeichnen und Parameter an Unterprogramme übergeben

Ein Blatt



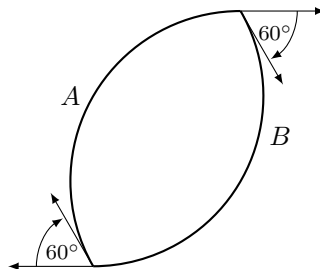
kann man als zwei zusammengeklebte Teilkreise *A* und *B* ansehen. Einen Teilkreis kann man zum Beispiel mit folgendem Programm zeichnen:

```
repeat 120 [fd 3 rt 1]
```

Probiere es aus.

Wir sehen, dass dieses Programm sehr ähnlich zu dem Programm für den Kreis ist. Anstatt 360 kleinen Schritten mit jeweils  $1^\circ$  Drehung, machen wir nur 120 kleine Schritte [fd 3 rt 1], und zeichnen dadurch nur ein Drittel des Kreises ( $120^\circ$ ).

Jetzt ist die Frage, wie viel man die Schildkröte drehen muss, bevor man den Teilkreis *B* für die untere Seite des Blattes zeichnet. Schauen wir uns das auf dem folgenden Bild an:



Wenn wir am Ende die ursprüngliche Position erreichen wollen, müssen wir die Schildkröte insgesamt wie immer um  $360^\circ$  drehen. Im Teil *A* drehen wir sie um  $120^\circ$  und im Teil *B* ebenfalls um  $120^\circ$ . Dann bleiben also noch

$$360^\circ - 120^\circ - 120^\circ = 120^\circ$$

übrig, die man gleichmässig auf die zwei Drehungen an den Spitzen des Blattes verteilen muss:

$$\frac{120^\circ}{2} = 60^\circ.$$

Damit erhalten wir folgendes Programm:

```
repeat 120 [fd 3 rt 1]
rt 60
repeat 120 [fd 3 rt 1]
rt 60
```

oder noch einfacher:

```
repeat 2 [repeat 120 [fd 3 rt 1] rt 60]
```

Probiere es aus.

Jetzt könnten wir uns wünschen, schmalere Blätter (die Teile *A* und *B* sind kürzer) oder breitere Blätter (die Teile *A* und *B* sind länger) zu zeichnen. Dazu kann man wieder einen Parameter verwenden. Nennen wir den Parameter zum Beispiel **:LANG** (wie Länge). Dann berechnen wir die Drehung an der Spitze des Blattes wie folgt:

Bevor man den Teil *B* des Blattes zeichnet, soll die Hälfte der ganzen Drehung, das heisst  $\frac{360^\circ}{2} = 180^\circ$ , gemacht werden. Also ist die Drehung an der Spitze des Blattes

$$180^\circ - \text{:LANG}.$$


Damit können wir folgendes Programm im Editor schreiben:

```
to BLATT :LANG
repeat 2 [repeat :LANG [fd 3 rt 1] rt 180-:LANG]
end
```

Probiere dann das Programm aus, indem du die folgenden Aufrufe in die Befehlszeile schreibst:

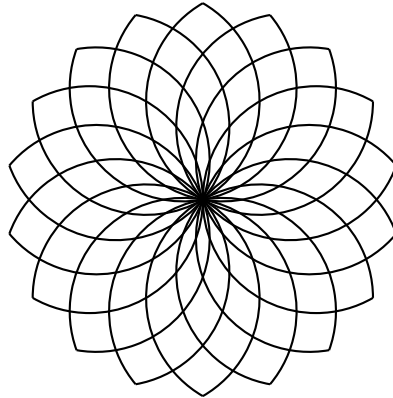
```
BLATT 20
BLATT 40
BLATT 60
BLATT 80
BLATT 100
```

Was passiert?

 **Aufgabe 56** Fantasiemuster


Zeichne zuerst mit folgendem Programm eine Blume:

```
BLATT 100  
rt 20  
BLATT 100  
rt 20  
BLATT 100  
....
```



Wie oft musst du die Befehle **BLATT** und **rt 20** wiederholen, um diese Blume vollständig zu zeichnen?

Schreibe das Programm für die Blume nur in einer Zeile mit einem geeigneten **repeat**-Befehl. (Denke dabei daran, dass alle Drehungen **rt** zwischen den einzelnen Blättern zusammen  $360^\circ$  ergeben müssen.)

 **Aufgabe 57** Im Programm **BLATT** bestimmt der Befehl **fd 3** die Grösse des Kreises, aus dem wir den Teilkreis der Länge **:LANG** schneiden. Diesen Wert **3** können wir auch durch einen Parameter namens **:GR** (Grösse) ersetzen. Schreibe ein Programm

```
BLAETTER :LANG :GR
```

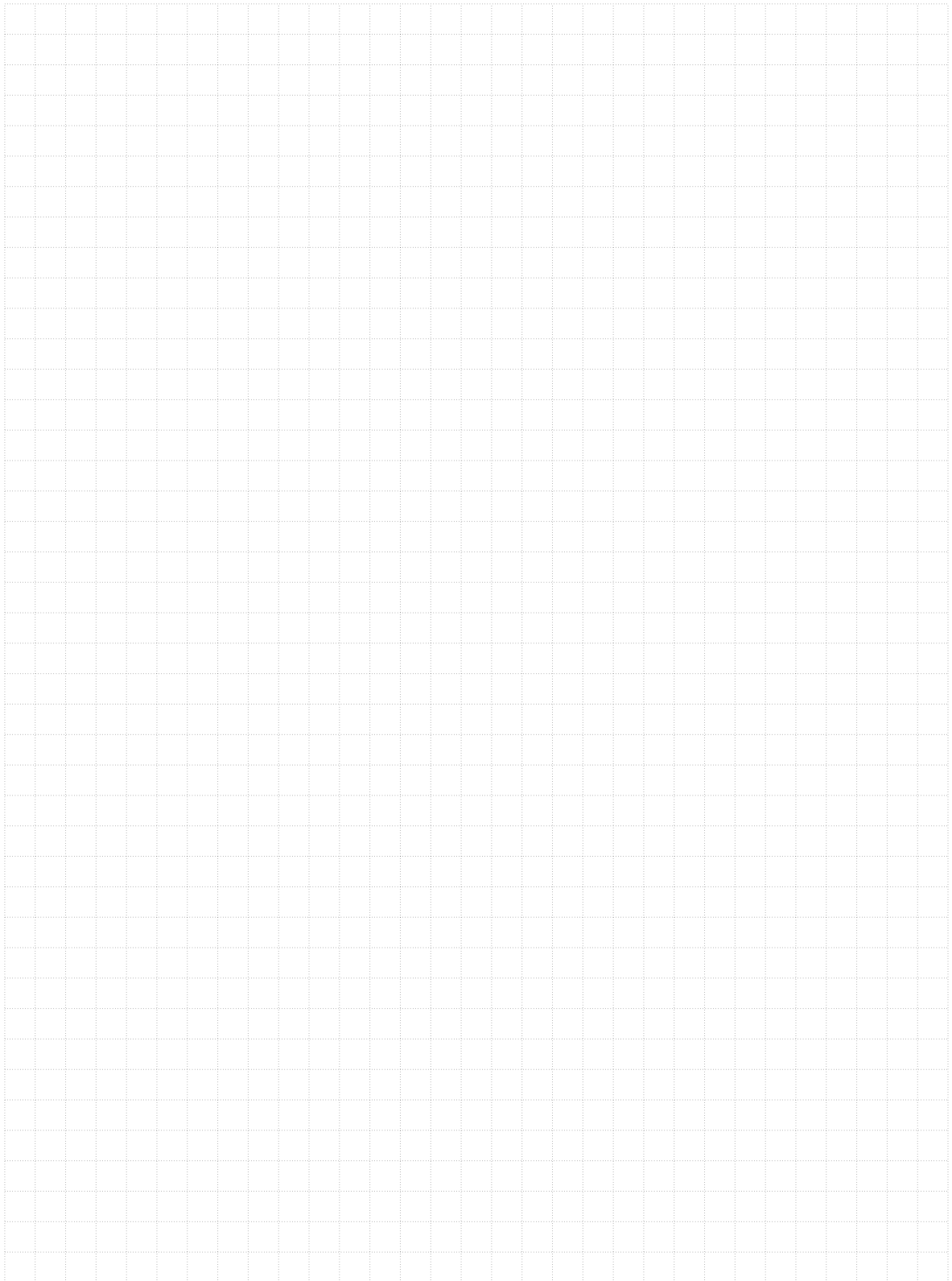
mit den Parametern **:LANG** und **:GR**, mit denen wir beides, die Länge und die Grösse, einstellen können. Probiere es dann aus mit den folgenden Programmaufrufen:

```
BLAETTER 100 1  
BLAETTER 100 1.5  
BLAETTER 80 2  
BLAETTER 80 2.5  
BLAETTER 70 3
```

Drehe dann die Schildkröte um  $180^\circ$  und wiederhole das Gleiche.

 **Aufgabe 58** Denke dir andere Fantasiebilder aus.

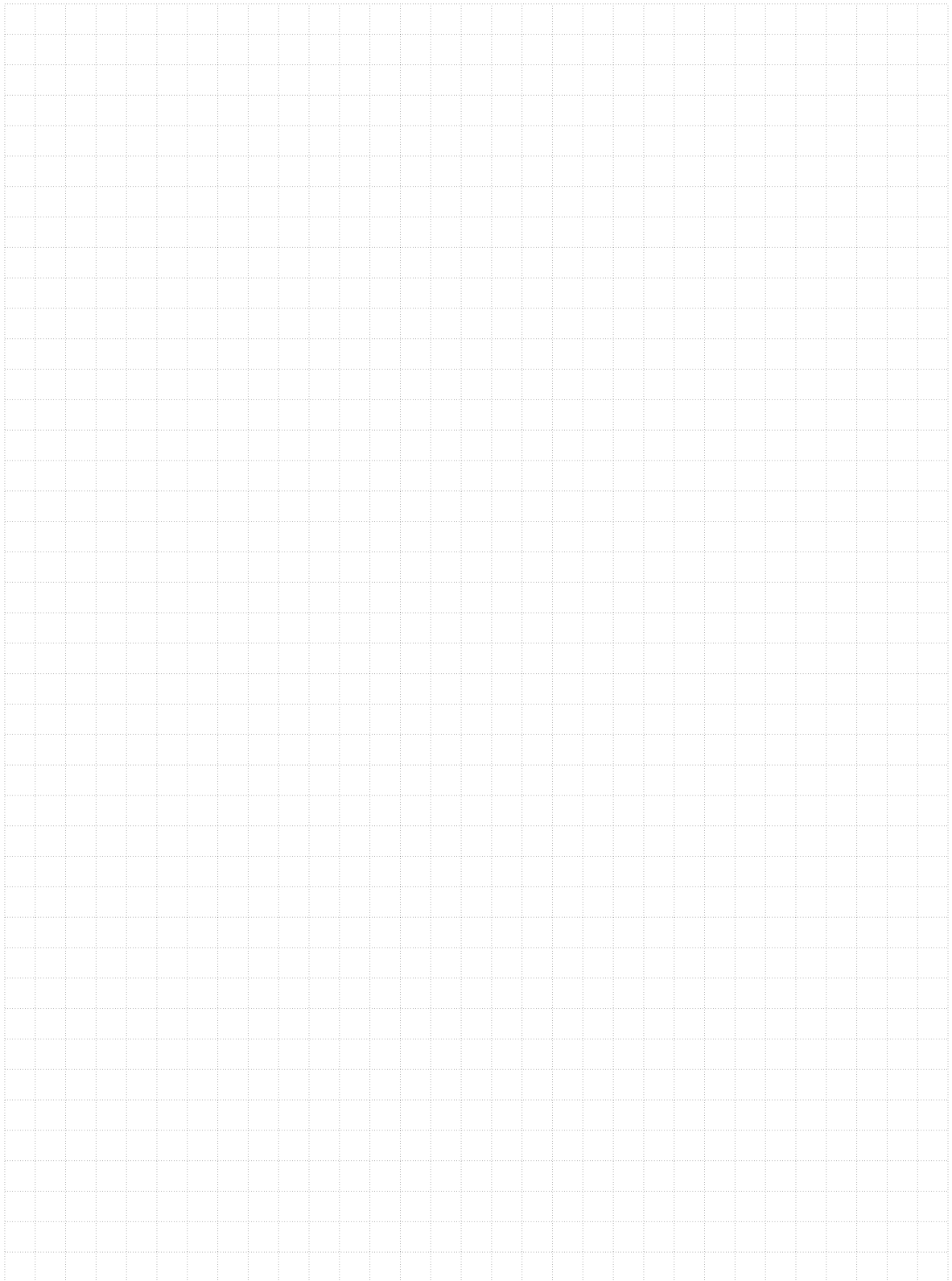
## Meine Notizen



# Meine Notizen



## Meine Notizen



```

to SCHILDKROETE
ht setpw 3
#Panzer
setpc 11
repeat 180 [rt 1 fd 3]
rt 80 repeat 20 [rt 1 fd 17.2]
home
#Kopf
setpc 16
repeat 20 [rt 1 fd 3]
lt 50 fd 30 rt 30
repeat 180 [lt 1 fd 0.7]
lt 90 pu fd 15 lt 90 pd
setpc 0
fd 15 repeat 180 [rt 1 fd 0.2] fd 15 rt 90 fd 180*0.2*2/pi
rt 90 repeat 6 [fd 8 rt 90 fd 1 rt 90 fd 8 rt 180] lt 90 fd 6
pu fd 15 pd
setpc 16
rt 30 repeat 220 [lt 1 fd 1] repeat 40 [bk 1 rt 1]
repeat 100 [lt 1 fd 1.2]
rt 90 repeat 10 [fd 0.5 lt 1] repeat 20 [rt 1 bk 0.5] repeat 10 [fd 0.5 rt 1] lt 120
repeat 40 [bk 1.2 rt 1] rt 160
home setpc 11
#Schwanz
repeat 180 [rt 1 fd 3]
setpc 16
lt 110 fd 50 lt 150 fd 50
setpc 11 pu home pd
#Füsse
repeat 180 [rt 1 fd 3]
rt 80 repeat 4 [rt 1 fd 17.2]
setpc 16
lt 90 fd 40 repeat 180 [rt 1 fd 30*pi/360] fd 40
setpc 11 pu home pd
repeat 180 [rt 1 fd 3]
setpc 16
rt 80 repeat 15 [rt 1 fd 17.2]
lt 90 fd 40 repeat 180 [rt 1 fd 30*pi/360] fd 40
setpc 11 pu home pd
repeat 180 [rt 1 fd 3]
rt 80 repeat 3 [rt 1 fd 17.2]
setpc 16
lt 140 fd 30 repeat 180 [rt 1 fd 30*pi/360] fd 25
setpc 11 pu home pd
repeat 180 [rt 1 fd 3]
rt 80 repeat 14 [rt 1 fd 17.2]
setpc 16
lt 140 fd 30 repeat 180 [rt 1 fd 30*pi/360] fd 25
setpc 11 pu home pd
#Flecken
pu fd 20 rt 90 fd 10 lt 90 pd
repeat 5 [FLECK 0.5]
pu rt 90 fd 40 lt 90 pd]
pu home fd 95 rt 90 fd 60 lt 90 pd FLECK 0.8
pu rt 90 fd 90 lt 90 pd FLECK 0.8
end

to FLECK :GR
repeat 8 [repeat 180 [fd :GR rt 1] make "gr 0.8*:GR]
end

```

Informationstechnologie und Ausbildung  
ETH Zürich, CAB F 15.1  
Universitätstrasse 6  
CH-8092 Zürich

[www.ite.ethz.ch](http://www.ite.ethz.ch)  
[www.abz.inf.ethz.ch](http://www.abz.inf.ethz.ch)