

## Lektion 7: Prozeduren, die sich selbst aufrufen

Bearbeitet von Daniel Labas, Atal Ashna, Jens Haendeler

In den bisherigen Abschnitten haben wir gelernt, dass Prozeduren in anderen Prozeduren als Befehle enthalten sein können. In der Sprache LOGO können Prozeduren sogar sich selbst als Befehle enthalten, also sich selbst aufrufen.

### Problem 7.1

Schreibe eine Prozedur für folgende Turtlebewegung: Die Turtle soll sich längs der Seiten eines Quadrats bewegen. Er zeichnet also fortwährend ein Quadrat.

Brich die Programmausführung mit dem Befehl **HALT** im Commander-Menue ab. Was bedeutet der Kommentar, den der Computer gibt? Wie kann man diese Vermutung bestätigen?

### Lösung

Damit die Turtle fortwährend den vorgegebenen Weg entlang läuft, muss ihr ständig der Befehl `fd :seite rt 90` gegeben werden. Statt des `repeat`-Befehls übernehmen wir die Prozedur selbst als Befehl in die Prozedur:

```
to quadratweg :seite  
fd :seite rt 90  
quadratweg :seite  
end
```

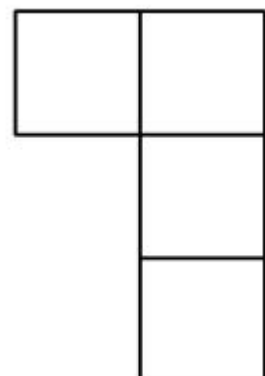


Unterbrechen wir die Durchführung der Prozedur mit dem **HALT**-Befehl, dann meldet der Computer z. B. *Stopping... in quadratweg*

### Probiere aus ...

- a) Wenn wir den Programmablauf nicht durch **HALT** unterbrechen, läuft die Prozedur immer weiter.

Man kann dann trotzdem neue Befehle eingeben, die der Rechner zwischendurch abarbeitet. So kann es z. B. geschehen, dass nach Eingabe des Befehls `quadratweg 100` und dreifachem Eingeben des Befehls `fd 100` der Rechner nahezu zufällig diese drei Vorwärtsbewegungen einschleibt und die rechts abgebildete Grafik erzeugt. Es kann auch eine andere Grafik entstehen.

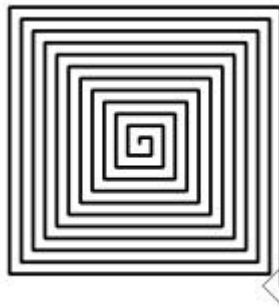


**Achtung:** Es kann leicht geschehen, dass das System „zusammenbricht“, wenn man mehrere sich selbst aufrufende Prozeduren gleichzeitig laufen lässt ...

- b) Man kann die Halt-Taste als Stopp-Mechanismus für ein Wettspiel mit der Prozedur *quadratweg* benutzen. Wo bleibt die Turtle stehen? Schafft man es, die Turtle in einer bestimmten Ecke anzuhalten?
- c) Man kann den Ablauf einer sich selbst aufrufenden Prozedur durch den Befehl *wait* verlangsamen. Nach dem Prozedurnamen *wait* muss eine positive Zahl eingegeben werden; *wait 60* bedeutet, dass das Abarbeiten der Befehle um 1 Sekunde aufgehalten wird.

## Problem 7.2

Zeichne eine rechteckige Spirale, bei der die erste Strecke 5 Einheiten lang ist, die zweite 10, die dritte 15, die vierte 20 usw. Damit man das entstehende Bild überhaupt erkennen kann, sollte der Ablauf mithilfe des *wait* – Befehls verlangsamt werden.



## Lösung

Ähnlich wie bei der Prozedur *quadratweg* können wir auch hier in der Prozedur die Prozedur selbst aufrufen, nur mit dem Unterschied, dass wir gegenüber dem letzten Prozeduraufruf den Wert des Parameters um 5 erhöhen:

```
to quadratweg :seite
fd :seite rt 90 wait 12
quadratweg :seite + 5
end
```

Die Prozedur muss mit einem konkreten Parameterwert gestartet werden, also hier mit *quadratweg 5*. Diese Prozedur ruft dann die Prozedur *quadratweg 10* auf, diese wiederum *quadratweg 15* usw.

Prozeduren, die sich selbst aufrufen, nennt man **rekursive Prozeduren**. Beim Wiederaufruf der Prozedur kann man den Parameterwert beibehalten (vgl. Problem 9) oder verändern (vgl. Problem 10).

Unbefriedigend ist es, dass mit den bisherigen Kenntnissen der Programmablauf nur durch Abbruch beendet werden kann. In der folgenden **Lektion 8** wird erläutert, wie man durch Bedingungen den Programmablauf beendet.

## Überlege ...

- a) Jemand möchte das Quadrat aus Problem 9 symmetrisch zur Mitte des Bildschirms zeichnen. Warum funktioniert die folgende Prozedur nicht wie gewünscht?
- b) In der Prozedur *spirale* muss man für den Parameter *seite* ein Startwert eingeben. Überlege, wie man dies vermeiden kann. Schreibe dazu eine Prozedur *start* (ohne Parameter), die dann die Prozedur *spirale* mit dem notwendigen Anfangswert aufruft.

## Übungen

### Ü7.1

- a) Schreibe eine Prozedur analog zu *spirale*, bei der sich die Seitenlänge fortlaufend verkleinert. Was geschieht, wenn die Seitenlänge 0 unterschritten wird?
- b) Mit Hilfe der Prozedur *sternpoly* kann man so genannte Sternpolygone zeichnen:

Für welche Winkel ergeben sich regelmäßige Stern-Ecke? Gibt es zu einer vorgegebenen Anzahl von Stern-Ecken (Zacken) auch immer passende Werte für den *winkel*? Für welche Anzahl von Ecken (Zacken) gibt es unterschiedliche Figuren?

### Ü7.2

Schreibe eine rekursive Prozedur zum fortlaufenden Zeichnen

- a) eines Rechtecks,
- b) eines gleichseitigen Dreiecks,
- c) eines Kreises,
- d) eines n-Ecks.

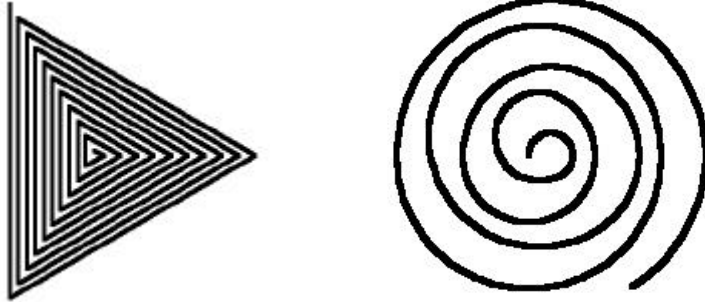
### Ü7.3

In der Prozedur *spirale* wurde die Seitenlänge um einen konstanten Betrag erhöht. Verändere die Seitenlänge mit einem konstanten Faktor.

### Ü7.4

Schreibe eine Prozedur, durch die analog zur obigen Aufgabe eine ‚Spirale‘ gezeichnet wird

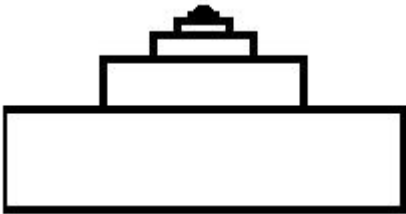
- a) in dreieckiger Form mit zunehmender Seitenlänge,
- b) durch Aneinandersetzen von Halbkreisen mit zunehmendem Radius.



### Ü7.5

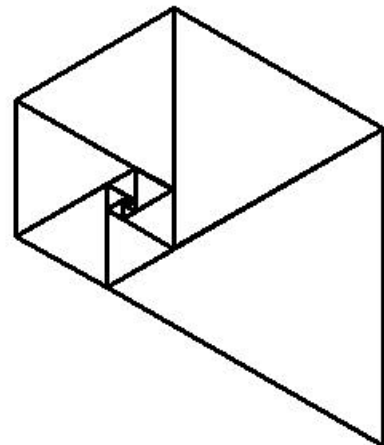
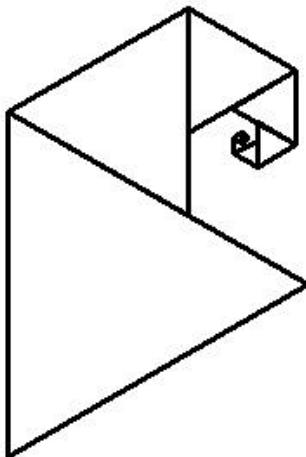
Schreibe eine Prozedur, mit deren Hilfe immer kleiner werdende Rechtecke aufeinander gesetzt werden.

*Beispiel: Das unterste Rechteck hat die Seitenlängen 120 und 200, das darüber liegende 60 und 100, dann 30 und 50 usw.*



### Ü7.6

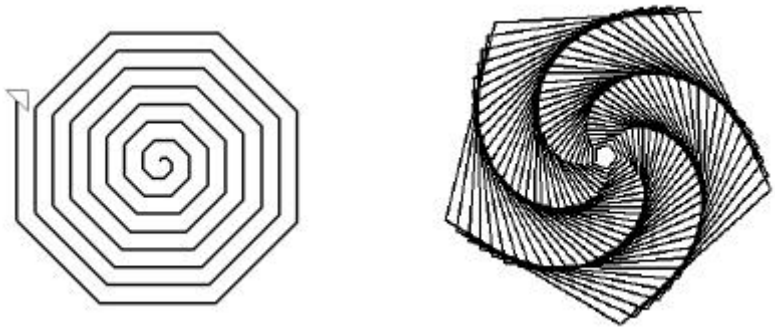
Schreibe eine Prozedur, mit der eine ‚Spirale‘ von aneinandergesetzten gleichseitigen Dreiecken gezeichnet wird, deren Seitenlänge abnimmt (zunimmt).



### Ü7.7

Schreibe eine Prozedur, durch die eine n-eckige ‚Spirale‘ gezeichnet wird, mit variabler Eingabe der Anfangslänge der Seite und des Drehwinkels.

- a) Die Seitenlänge werde um einen festen Betrag vergrößert (Parameter *zuwachs*).  
 Probiere Winkel aus, die nahe bei dem Drehwinkel eines regelmäßigen n-Ecks liegen.



b) Der Winkel werde um einen festen Betrag vergrößert. Für welche Winkelzuwächse entstehen regelmäßige Figuren?

