5. Befehle mit Parametern

Lernziele:

- Du lernst, wie man einen Befehl mit Hilfe von Parametern verallgemeinern kann.
- Du lernst, wie die Übergabe von Parametern zwischen Befehlen funktioniert.

5.1 Befehle verallgemeinern mit Parametern

Die Ausgabe der drei Befehle unterscheidet sich lediglich in der Seitenlänge der Sternfigur.

```
1 def stern1():
2     rt(90)
3     repeat 5:
4         fd(50)
          rt(144)
```

```
1 def stern2():
2     rt(90)
3     repeat 5:
4         fd(100)
          rt(144)
```

```
1 def stern3():
2     rt(90)
3     repeat 5:
4         fd(200)
          rt(144)
```

Beispiel 5.1:

Anstatt drei Befehle zu definieren, möchten wir nur noch einen Befehl, dem wir die Seitenlänge als Parameterwert übergeben können: `stern(50)` oder `stern(100)` oder `stern(200)`

Dazu müssen wir bei der Befehlsdefinition in Klammern einen allgemeinen Parameter definieren:

```
1  from gturtle import*
2
3  def stern(seite):
4      rt(90)
5      repeat 5:
6          fd(seite)
7          rt(144)
8
9  ###################
10 makeTurtle()
11
12 stern(100)
```

`seite` wird hier als Parameter des Befehls festgelegt

Bei der Ausführung von `stern(100)` setzt der Computer für `seite` den Wert `100` ein.

Ausführung des Befehls `stern(100)`

Die Bezeichnung für einen Parameter kann frei gewählt werden, man sollte aber immer aussagekräftige Bezeichnungen wählen (wie hier `seite`).

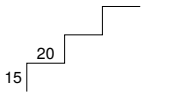
Wichtig: Sobald man einen Befehl mit einem Parameter definiert hat, erwartet dieser beim Aufruf einen Wert und es erscheint eine Fehlermeldung, falls in den Klammern kein Wert übergeben wird.

Begriffe

Parameter (auch **Argumente** genannt) sind Werte, welche man Befehlen bei der Ausführung in den runden Klammern übergeben kann. Damit kann ein Befehl, abhängig von den Parameterwerten, unterschiedliche Tätigkeiten ausführen.

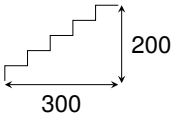
Aufgabe 5.1

(a) Definiere einen Befehl `treppe(n)`, der eine Treppe mit `n` Stufen der Breite 20 und Höhe 15 zeichnet.

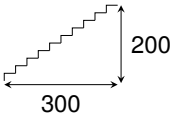


`treppe(3)`

(b) Eine Treppe ist 300 Pixel breit und 200 Pixel hoch. Definiere `treppe(n)` so, dass die Länge und Höhe der Stufen in Abhängigkeit der Anzahl `n` berechnet und gezeichnet werden.



`treppe(5)`



`treppe(10)`

5.2 Befehle mit mehreren Parametern

Beispiel 5.2

Einem Befehl kann man mehrere Parameter übergeben, die jeweils durch ein Komma getrennt werden müssen. `quadrat(seite,farbe,dicke)` hat drei Parameter für die Seitenlänge, die Stiftfarbe und die Stiftdicke.

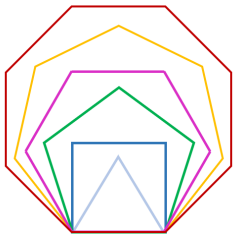
```
1  from gturtle import*
2
3  def quadrat(seite,farbe,dicke):
4      setPenWidth(dicke)
5      setPenColor(farbe)
6      repeat 4:
7          fd(seite)
8          rt(90)
9
10 ###################
11 makeTurtle()
12 ht()
13
14 quadrat(100,"orange",5)
```

`quadrat(100,"orange",5)`

Aufgabe 5.2

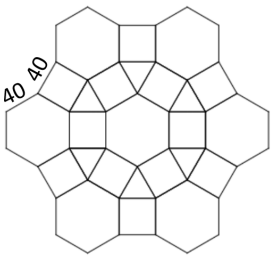
Definiere einen Befehl `vieleck(n,seite,farbe,dicke)`, der ein regelmässiges `n`-Eck mit der Seitenlänge `seite`, der Farbe `farbe` und der Stiftdicke `dicke` zeichnet.

Erstelle damit die abgebildete Folge von Vielecken mit aufsteigender Eckenzahl.



Aufgabe 5.3

Definiere einen Befehl `vieleck(n,seite)` und erstelle damit die folgende Parkettierung mit möglichst wenig Zeilen.



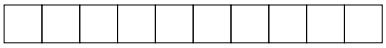
5.3 Übertragung von Parameterwerten zwischen Befehlen

Verwendet man im Körper eines Befehls wiederum Befehle als Unterprogramm, so kann man die Parameterwerte des Hauptprogramms an die Parameter des Unterprogramms übertragen:

Beispiel 5.3

Der Befehl `reihe(n,seite)` soll `n` Quadrate mit der Seitenlänge `seite` horizontal nebeneinander zeichnen. Im Körper des Befehls wird der Parameterwert von `seite` an den Parameter `seite` des Befehls `quadrat(seite)` übertragen:

```
1  from gturtle import*
2
3  def quadrat(seite):
4      repeat 4:
5          fd(seite)
6          rt(90)
7
8  def reihe(n, seite):
9      repeat n:
10         quadrat(seite)
11         rt(90)
12         fd(seite)
13         lt(90)
14
15 ####################
16 makeTurtle()
17 ht()
18
19 reihe(10,30)
```

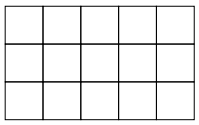


reihe(10,30)

Die Bezeichnung des Parameters `seite` ist in beiden Befehlen gleich, was nicht zwingend so sein muss. Es ist hier aber sinnvoll, da sie den exakt gleichen Wert (Quadratseite) darstellen.

Aufgabe 5.4

Ergänze das Programm aus Beispiel 5.3 mit einem Befehl `gitter(m,n,seite)` der ein m×n-Gitter (m Zeilen und n Spalten) aus Quadraten mit der Seitenlänge `seite` zeichnet.



gitter(3,5,30)