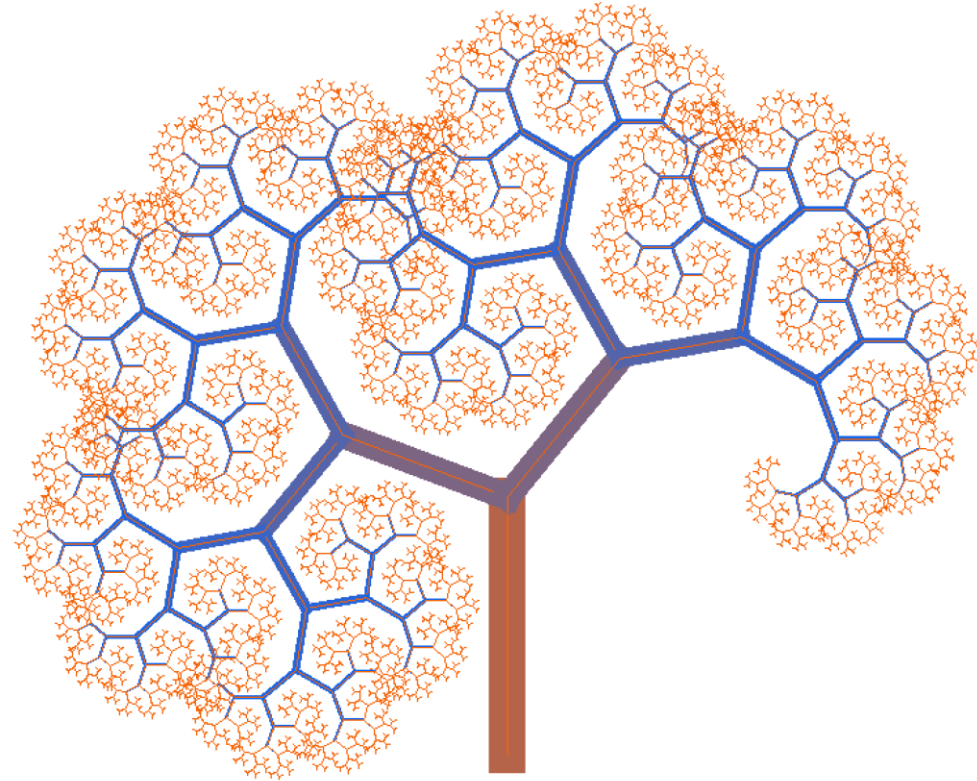


Programmieren mit TigerJython



Einstiegsübung 3

Angenommen, du hättest den Auftrag, Programme für die Grafiken (a) bis (d) zu entwickeln.

- Du möchtest so effizient arbeiten wie möglich.
- Du arbeitest in einem Team und euer Code soll für die anderen gut lesbar/erweiterbar sein.

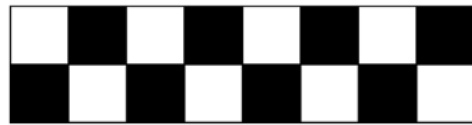
Wie kannst du deine Programme gestalten, damit du diese beiden Ziele erreichst?
(Besprecht euch in einem **2er-Team**)



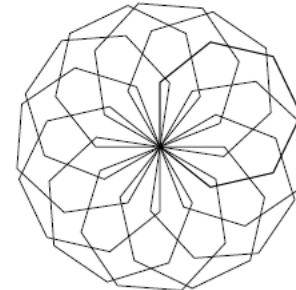
(a)



(b)



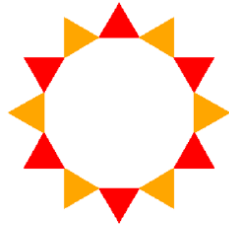
(c)



(d)

Einstiegsübung 3

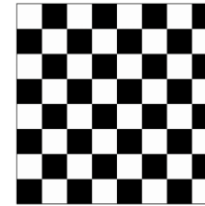
Die Grafiken der Einstiegsübung sind aus identischen **Bausteinen** (Vielecke, Sterne) aufgebaut.



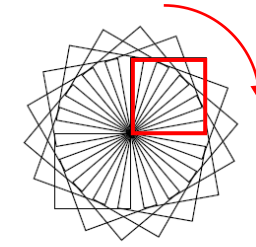
(a)



(b)



(c)



(d)

Bausteine:



Bausteine nicht immer neu programmieren, sondern einmal erstellen und im Programm verwenden.

Ähnlich wie ein Stempel, den man einmal herstellt und dann immer wieder benutzt.



3. Der Modulare Entwurf

Lernziele:

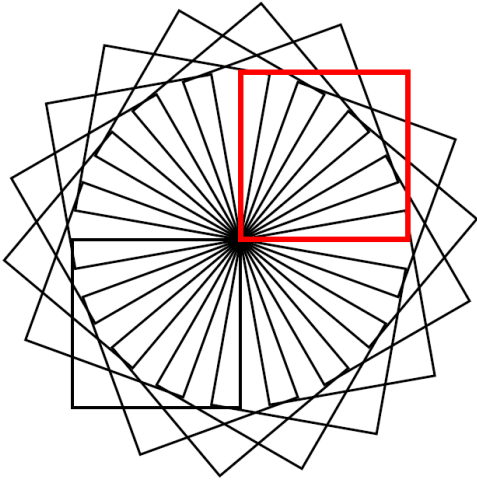
Theorie

- Du kennst den **modularen Entwurf** und weisst wie man damit strukturiert programmieren kann.
- Du weisst, wie man mit eigenen **Befehlen** die Programmiersprache erweitert.

Praxis

- Du lernst, wie du eigene **Befehle definieren** und **verwenden** kannst.
- Du weisst, wie man beim Programmieren eine Problemstellung in kleine Bausteine zerlegt.

Neue Begriffe

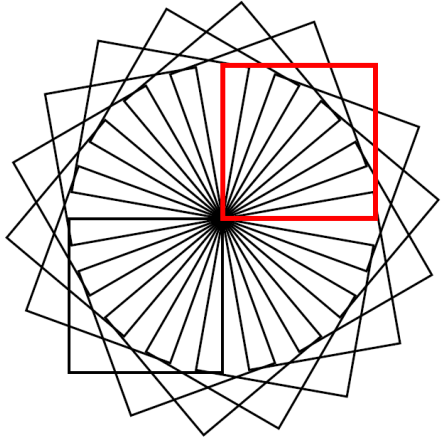


```
1 from turtle import *
2
3 def quadrat():
4     repeat 4:
5         fd(20)
6         rt(90)
7
8 #####
9 makeTurtle()
10 ht()
11
12 repeat 18:
13     quadrat()
14     rt(20)
```

← Mit **def** wird ein neuer **Befehl** definiert. Den Namen kannst du selber wählen. Wichtig sind die runden Klammern und der Doppelpunkt.

← Der **Körper des Befehls** wird eingerückt. Er beschreibt, was ausgeführt werden soll, wenn der Befehl aufgerufen wird.

← **Aufruf des Befehls**. Erst jetzt zeichnet der Computer.



```
from turtle import *
```

Hauptprogramm

```
def quadrat():  
    repeat 4:  
        fd(20)  
        rt(90)
```

Unterprogramm

```
#####
```

```
makeTurtle()  
ht()
```

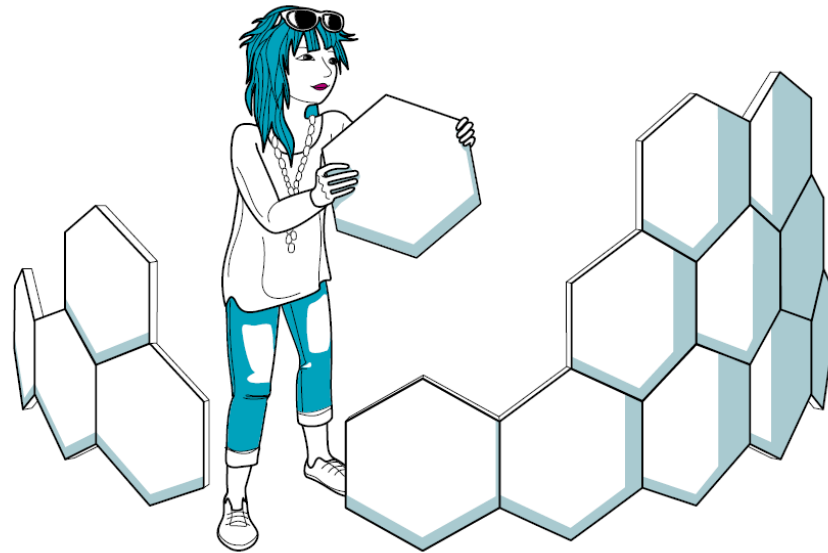
```
repeat 18:  
    quadrat()  
    rt(20)
```

← *Befehl aufrufen*

Neue Begriffe

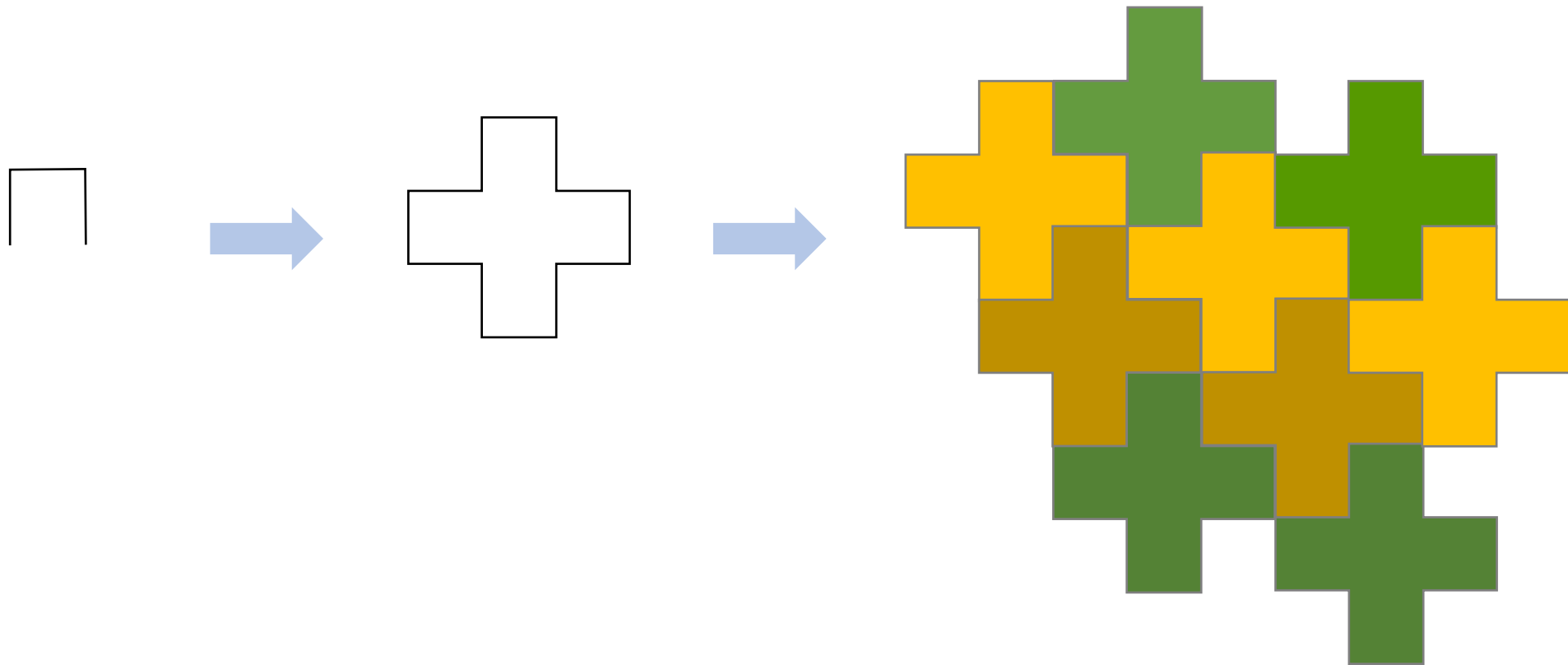
Beim **modularen Entwurf** handelt es sich um Prinzip zum Bau von komplexen Systemen.

Beim Programmieren wenden wir es an, in dem wir kleine Programmbausteine als **Befehle** definieren und diese benutzen, um komplexere Programme zu bauen, aus denen wiederum noch komplexere Programme erstellt werden können.



Beispiel

In Beispiel 3.2 ist ein Kreuz aus vier kleinen Einzelteilen gebaut worden. Das Kreuz selber kann dann wiederum ein Modul eines noch komplexeren Programms sein, wie hier die Parkettierung:



Bearbeite die Aufgaben **3.1 bis 3.8** am Computer



Aufgabe 3.4

Lösung:

Zeichne die Ausgabe des Programms ins Gitter.

```
1 from turtle import *
2
3 def teil():
4     repeat 3:
5         fd(30)
6         rt(90)
7     lt(180)
8
9 #####
10
11 makeTurtle()
12 ht()
13 repeat 4:
14     teil()
```

