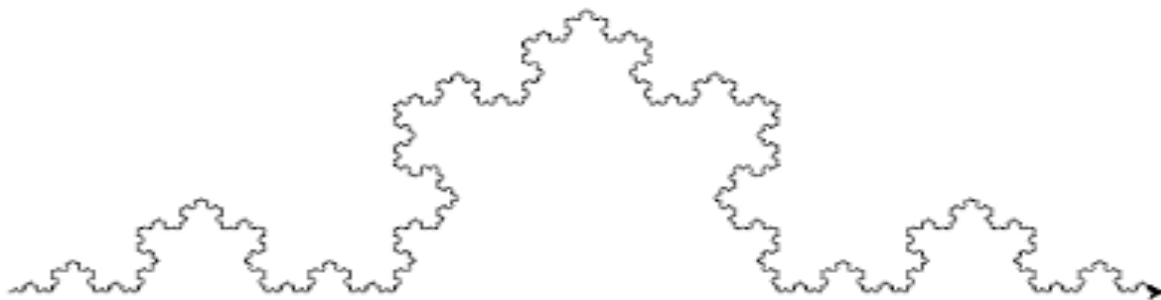


Rekursion mit Python-Turtle

1 Einführung

Die Python-Turtle ist eine Bibliothek von Python, die es ermöglicht, mit Befehlen eine kleine Figur über den Bildschirm laufen zu lassen. Die Turtle kann ihren Weg mit verschiedenen Farben und Stiften aufzeichnen und verschiedene komplexe Zeichnungen anfertigen. So lernt man leicht programmieren. Du findest im Internet viele Beispiele für das Programmieren mit der Turtle. Einen Überblick über die wichtigsten Befehle für die Turtle findest du unter <http://python4kids.net/>



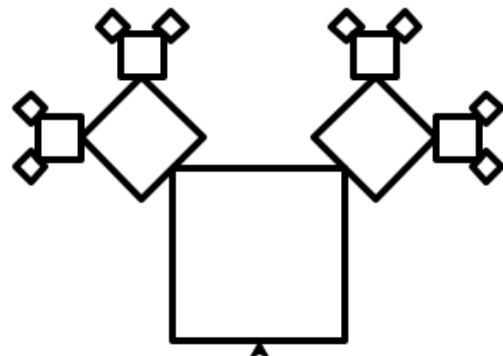
2 Rekursion am Beispiel des Quadratbaumes

Als erstes Beispiel soll eine rekursive Struktur gezeichnet werden. Es ist eine einfache geometrische Figur, die sich in unterschiedlicher Größe immer wiederholt. Man nennt solche Figuren auch selbstähnlich oder fraktale Figuren. Bei den sich wiederholenden Elementen kann sich aber auch eine Eigenschaft ändern. In unserem Fall wird die Seitenlänge bei jedem Quadrat halbiert. Das Zeichnen eines Quadrates ist nicht schwer. Am besten du verwendest gleich eine eigene Funktion und übergibst der Funktion einen Parameter für die Seitenlänge. So kannst du verschieden große Quadrate zeichnen.

```
1 from turtle import *
2
3 def zeichne_quadrat(wert):
4     for i in range(4):
5         forward(wert)
6         right(90)
7
8 #Hauptprogramm
9 reset()
10 pensize(4)
11 shapesize(stretch_wid=2, stretch_len=2, outline=None)
12 zeichne_quadrat(100)
13 done()
```

Die Lösung für einen Quadratbaum sieht nun so aus:

```
1 from turtle import *
2
3 def zeichne_quadratbaum(wert):
4     if (wert < 100/2**((zahl-1))):
5         return
6     else:
7         left(90)
8         forward(wert/2)
9         right(90)
10        forward(wert)
11        left(45)
12        zeichne_quadratbaum(wert/2)
13        right(135)
14        forward(wert)
15        left(45)
16        zeichne_quadratbaum(wert/2)
17        right(135)
18        forward(wert)
19        right(90)
20        forward(wert/2)
21        right(90)
22
23 #Hauptprogramm
24 left(90)
25 zahl=int(input("Rekursionstiefe
26                eingeben: "))
27 zeichne_quadratbaum(100)
28 done()
```



Quadratbaum der Tiefe 4

Listing 1: Quadratbaum.py

Wie kann man diese Lösung finden? Ihr versucht sicherlich zuerst mit Schleifen die Lösung zu beschreiben. Das gibt aber Probleme. Die Lösung ist sehr einfach darstellbar, wenn man Rekursion verwendet. Betrachten wir dazu den Quelltext genauer. In der Funktion `zeichne_quadratbaum(100)` gibt der übergebene Parameter die Kantenlänge des zu zeichnenden Quadrates an. Sinkt der Wert unterhalb einer vorgegebenen Grenze, so ist der Rekursionsanker erreicht. Die Variable `zahl` gibt die Anzahl der rekursiven Durchläufe an und bestimmt die Abbruchbedingung.

Unten sind die Programmzeilen 7 bis 11 von Listing 1 dargestellt. Die Turtle ist standardmäßig nach rechts ausgerichtet. Daher erfolgt im Hauptprogramm einmalig die Drehung nach links (Zeile 24 Listing 1). Dann geht das Zeichnen des halben Quadrates los. An der Ecke stoppt die Turtle und dreht sich nach links. Dann ist die gleiche Ausgangssituation erreicht wie anfangs. Es erfolgt nun der rekursive Aufruf mit `zeichne_quadratbaum(50)`.

```
7     left (90)
8     forward ( wert / 2)
9     right (90)
10    forward ( wert )
11    left (45)
```



Wenn die Abbruchbedingung nicht erreicht ist, dann wird wiederum das halbe zweite Quadrat gezeichnet. An der zweiten Ecke wird das Zeichnen des zweiten Quadrates unterbrochen und wieder erfolgt der rekursive Aufruf mit `zeichne_quadratbaum(wert/2)`. Wenn dann der Rekursionsanker erreicht ist (Zeile 4 in Listing 1) so wird kein neuer Rekursionsaufruf gestartet. Die Funktion wird mit `return()` beendet. Die Turtle kann so den Rest des Quadrates fertigzeichnen (Abb. 3)



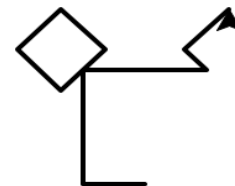
Abb 2.: Turtleposition vor Beginn der zweiten Rekursion



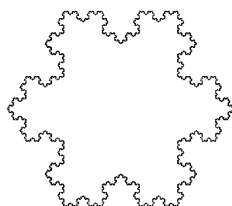
Abb 3.: Der Rekursionsanker ist erreicht, das kleine Quadrat ist fertig

Um nun an die andere Ecke zu kommen, werden die Befehle in Zeile 13 bis 15 ausgeführt. Dort wird dann wiederum rekursiv die Funktion `zeichne_quadratbaum(wert)` aufgerufen, die dann die Quadrate an der rechten Ecke rekursiv zeichnet.

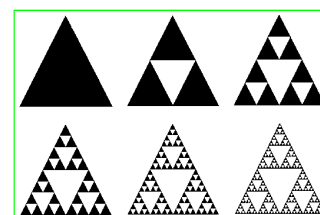
```
13    right (135)
14    forward ( wert )
15    left (45)
```



Aufgabe: Programmiere die Kochkurve (Schneeflockenkurve) und das Sierpinski-Dreieck.



Schneeflockenkurve



Sierpinski-Dreieck