## Beispiele aus dem Alltag

In der Literatur werden vielerlei Beispiele zur Veranschaulichung der Rekursion genannt. Darunter sind welche, bei denen die Komponente „Zeit“ keine Rolle spielt, d. h. der Selbstbezug auf den ersten Blick erkennbar ist. Es handelt sich hierbei etwa um Verpackungen, auf deren Etikett die gleiche Verpackung dargestellt ist, oder um ein Fernsehbild, das einen Ansager zeigt, neben der ein Fernseher steht, auf der man das Fernsehbild und damit den Ansager wieder sehen kann, usw. Auch ein Gegenstand, der zwischen zwei parallele Spiegel gehalten wird, erscheint vervielfacht.

## Rekursive Grafiken

Im Mathematikunterricht dominiert die Geometrie im Koordinatensystem. Leistungsfähige geometrische Algorithmen sind auch ohne Koordinatengeometrie möglich, wenn man eine *Turtle* einsetzt. Wir benutzen eine Turtle, die als selbstentwickelte Programmbibliothek in unser Programm eingefügt werden muss.

## Die Turtle

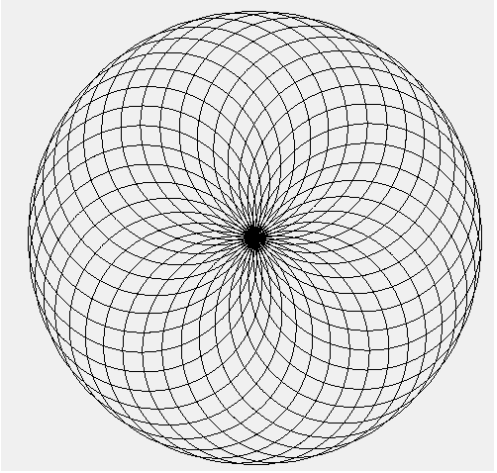
Anfangs befindet sich die Turtle in der Mitte der Zeichenfläche und schaut nach rechts. Methoden der Turtle die wir benötigen sind:

| **Methode** | **Beschreibung** |
|---|---|
| Mitte() | Setzt die Turtle in die Mitte der Zeichenfläche mit Blickrichtung nach rechts. |
| Vor(float *länge*) | Die Turtle zeichnet in der aktuellen Richtung eine *länge* Einheiten lange Strecke. |
| DreheRechts(float *winkel*) | Dreht die Zeichenrichtung der Turtle um *winkel* Grad nach rechts |
| DreheLinks(float *winkel*) | Dreht die Zeichenrichtung der Turtle um *winkel* Grad nach links |

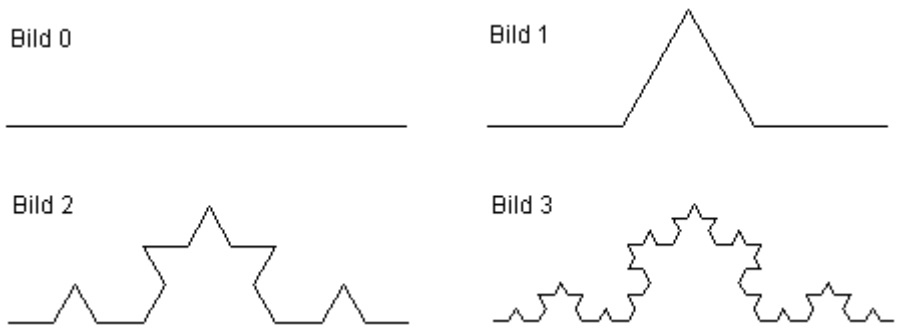
Damit das Turtleobjekt genutzt werden kann, muss dieses zuerst erzeugt und deklariert werden. Beispiel: `CTurtleGraphics myTurtle = new CTurtleGraphics(viewForm);` `viewForm` bezeichnet hier ein Objekt vom Typ Form auf welchem der Turtle malen soll. **Tipp:** Übergebe `this`, wenn du auf der gleichen Form malen willst.

### Aufgaben:

1. Lasse den Turtle eine Gerade zeichnen
2. Lasse den Turtle ein Viereck zeichnen
3. Lasse den Turtle einen Kreis zeichnen
4. Entwickle die Zeichenprozeduren für eine Blume (siehe Bild)



5. Entwickle die Zeichenprozeduren für die folgenden vier Bilder.




### Generator

Den charakteristischen *Generator* (Bild 1) erhält man so:

```
private void Kurve1(float laenge)
{
    myTurtle.Vor(laenge/3);
    myTurtle.DreheLinks(60);
    myTurtle.Vor(laenge/3);
    myTurtle.DreheRechts(120);
    myTurtle.Vor(laenge/3);
    myTurtle.DreheLinks(60);
    myTurtle.Vor(laenge/3);
}
```

### Streckenersetzung

Bild 2 erhält man am einfachsten durch Streckenersetzung aus Bild 1, in dem man jede Teilstrecke von Bild 1 durch den Generator ersetzt. In der Zeichenprozedur muss dazu jeder `vor`-Befehl durch den Aufruf der Zeichenprozedur für den Generator ersetzt werden. Damit die Größenverhältnisse stimmen muss die Länge gedrittelt werden.

```
private void Kurve2(float laenge)
{
    Kurve1(laenge / 3);
    myTurtle.DreheLinks(60);
    Kurve1(laenge / 3);
    myTurtle.DreheRechts(120);
    Kurve1(laenge / 3);
    myTurtle.DreheLinks(60);
    Kurve1(laenge / 3);
}
```

Bild 3 läßt sich wiederum mittels Streckenersetzung mit Hilfe der Zeichenprozedur von Bild 2 zeichnen:

```
private void Kurve3(float laenge)
{
    Kurve2(laenge / 3);
    myTurtle.DreheLinks(60);
    Kurve2(laenge / 3);
    myTurtle.DreheRechts(120);
    Kurve2(laenge / 3);
    myTurtle.DreheLinks(60);
    Kurve2(laenge / 3);
}
```

## Rekursion

Gesucht ist nun eine Prozedur, mit welcher man das Bild mit der Nummer n erzeugen kann.