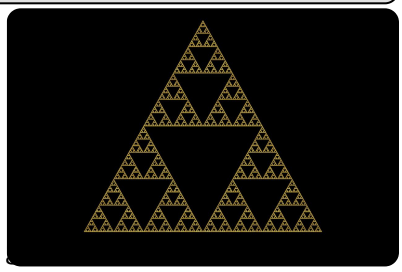
Rekursion ist neben der Iteration das mächtigste Werkzeug der Programmierung. Jedes denkbare Programm kann lediglich durch Zuweisungen, Verzweigungen und Rekursionen (oder Iterationen oder Sprünge) programmiert werden. Bevor wir uns der Rekursion zuwenden, betrachten wir **selbstähnliche Figuren: Fraktale**.

> **Fraktal**
>
> Ein **Fraktal** ist eine (geometrische) Figur, die aus einem **Teil** besteht, der **zu der gesamten Figur ähnlich** ist. Eine solche Figur muss zwangsweise unendlich sein, wir können sie nur annähern. Ein einfaches Beispiel ist das **Sierpinsky-Dreieck**.

① Sie benötigen ein leeres Blatt in DIN-A4-Größe, einen Bleistift und ein Lineal.

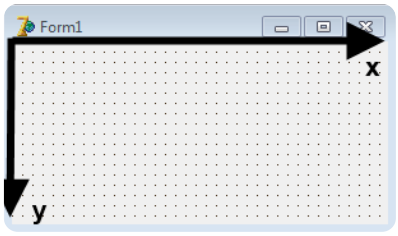
- Zeichnen Sie ein beliebiges Dreieck. Die Seitenlängen sollten ca. 10-15cm betragen.
- Markieren Sie die Mitten der Seiten und verbinden Sie jeweils zwei Seitenmitten miteinander durch eine Strecke. Es entstehen so 4 Teildreiecke.
- Wiederholen Sie den zweiten Schritt mit den drei äu-


Dreieck

Führt man dieses Verfahren **endlos** fort, so entsteht als Grenzfigur das so genannte **Sierpinsky-Dreieck**. Dabei ist jedes Teildreieck zum gesamten Dreieck **ähnlich**.

Das erste gezeichnete Dreieck nennt man **Initiator**. Er beschreibt das „initiale“ (erste) Objekt. Die Bildungsregeln 2 und 3 heißen **Generator**. Sie geben an, wie neue Dreiecke „generiert“ werden. Das folgende Arbeitsblatt begleitet die Umsetzung eines Programms zur Er-

Zeichnen mit Delphi



In jedem Formular in Delphi ist ein (ganzzahliges) Koordinatensystem eingebettet. Der Ursprung ist der Punkt links oben. Die x-Achse verläuft, wie üblich, von links nach rechts. **Die y-Achse dagegen wächst von oben nach unten**. Jedem Punkt auf dem Formular ist daher ein Paar ganzzahlig positiver Koordinaten zugeordnet. Die Einheit, in der die Koordinatenwerte angegeben sind, ist die Anzahl der Pixel, die der Punkt in waagerechter oder senkrechter Entfernung vom Ur-

Bewegen des Cursors

Die Fläche, auf der wir zeichnen, heißt **Canvas** (Leinwand) und ist ein Attribut des Formulars. Um den (unsichtbaren) Zeichencursor (ohne zu zeichnen) an eine bestimmte Position zu bringen, können wir die Operation **MoveTo** verwenden.
Beispiel: *Form1.canvas.MoveTo(23, 117);*
Der Befehl bewegt den Zeichencursor an den Punkt (23|117).

Zeichnen einer Strecke

Die Operation **LineTo** zeichnet eine Strecke vom aktuellen Punkt des Zeichencursors zu den Zielkoordinaten. Beispiel: *Form1.canvas.LineTo(47,11);*
Der Befehl zeichnet eine Linie von der aktuellen Position P zum Punkt (47|11).

Angaben zu den Urhebern und Lizenzbedingungen der einzelnen Bestandteile dieses Dokuments finden Sie unter
https://www.tutory.de/entdecken/dokument/36c7ef70

② (Vorbereitende Aufgabe)
Schreiben Sie ein Programm, das eine Strecke zwischen zwei vom Benutzer eingegebenen Punkten zeichnet. Die Eingabe der Koordinaten erfolgt für jeden Punkt durch Angabe der x- und der y-Koordinate in je einer Editkomponente.

③ Schreiben Sie eine Prozedur ZeichneStrecke, die eine Strecke zwischen zwei Punkten zeichnet. Verwenden Sie den vorgefertigten Datentyp TPoint. Testen Sie ihre Prozedur.

Datentyp TPoint

type TPoint = **record** x : integer; y : integer;**end**;

④ Zeichnen des Dreiecks

a) Der Benutzer gibt die Koordinaten dreier verschiedener Punkte A,B,C ein. Die Punkte sollen nicht auf einer Linie liegen. Das Programm zeichnet dann das Dreieck ΔABC.
b) Überlegen Sie sich mögliche Fehler, die dem Benutzer bei der Eingabe der Punkte unterlaufen könnten. Notieren Sie Beispiele und Lösungsmöglichkeiten.
c) Falsche oder unsinnige Eingaben sollen abgefangen werden. Der Benutzer erhält eine möglichst gute Fehlermeldung per showmessage-Befehl und die Werte werden

Da Fraktale jeweils nur geometrische Grenzwerte darstellen, können wir sie nur durch endlich viele Schritte annähern. Ein **Schritt** bedeutet, dass der **Generator einmal ausgeführt** wird. Unser Ziel ist nun, ein Programm zu schreiben, welches das Sierpinsky-Dreieck durch eine vom Benutzer eingegebene **Zahl von Schr**

Rechenweg

Die Koordinaten des Mittelpunktes M sind die Koordinaten des (gerundeten) arithmetischen Mittels der Koordinaten der Punkte A und B.
M.x=(A.x+B.x)/2

⑤ Schreiben Sie eine Funktion
Mittelpunkt: TPoint × TPoint → TPoint,
die den Mittelpunkt der Strecke zwischen zwei Punkten berechnet.

⑥ Sortieren Sie die Arbeitsschritte nach Ihrer Reihenfolge! (1-5)

- [ ] Berechne die Mittelpunkte der drei Seiten.
- [ ] Führe den Genrator mit den äußeren drei Teildreiecken aus.
- [ ] Verringere die Schrittzahl k um eins.
- [ ] Prüfe, ob die Zahl der noch auszuführenden Schritte k größer als null ist. Falls nein, beende das Zeichnen.
- [ ] Zeichne das Dreieck.

⑦ Schreiben Sie eine Prozedur Generator. Die Prozedur arbeitet die Schritte aus Aufgabe 6 nacheinander ab. Verwenden Sie folgende Signatur:
Procedure Generator(k: integer; A,B,C : TPoint);

Angaben zu den Urhebern und Lizenzbedingungen der einzelnen Bestandteile dieses Dokuments finden Sie unter
https://www.tutory.de/entdecken/dokument/36c7ef70