

TAREA 1

Copo de Nieve de Koch

Nombre Alumno : Gabriel Iturra Bocaz

Correo : gabrieliturab@ug.uchile.cl

Profesor : Patricio Poblete

Profesor Auxiliar : Manuel Cáceres
Sergio Peñafiel

Fecha : 31 de Marzo de 2016
Santiago, Chile.

Contenido

I.	Introducción	1
II.	Análisis del Problema	2
II.1.	Problema Propuesto	2
II.2.	Implementación	3
III.	Solución del Problema	4
III.1.	Desarrollo	4
III.2.	Definición de Variables	5
III.3.	Función curvaDeKoch	6
IV.	Ejemplos de Funcionamiento	8
V.	Modos de Uso	10
IV.	Resultados y Análisis	11

I. Introducción

El siguiente informe detalla el desarrollo de la Tarea 1 del curso Algoritmos y Estructuras de Datos (CC3001). Como primera tarea del curso, se pretende que el alumno se interiorice con el lenguaje de programación JAVA, familiarizándose con el manejo de Clases, tipos de variables, uso de funciones, booleanos, y condicionales.

Los fractales son figuras geométricas muy interesantes que parecen fragmentadas, tienen dimensión no entera y no pueden ser expresadas de una forma paramétrica. Además muchos de ellos tienen la propiedad de autosimilitud, es decir los fractales repiten su estructura a diferentes escalas. Este tipo de fractales se pueden construir a partir de la repetición de un patrón o regla.

Si bien no existe una ecuación que permita describir un fractal, desde el punto de vista computacional los fractales son simplemente la aplicación de una función de dibujo recursivamente. Por esto y para esta Tarea 1 se pidió modelar el copo de nieve de Koch, que es una curva cerrada no diferenciable en ningún punto descrita por matemático sueco Helge von Koch en 1904, en su artículo titulado “Acerca de una curva continua que no posee tangentes y obtenidas por los métodos de la geometría elemental”.

Para su desarrollo, se usó la Clase Turtle y StdDraw que a través de sus distintos métodos generaban el Canvas y las líneas encargadas de trazar las curvas de Koch, a través de la función recursiva llamada “curvaDeKoch” recibiendo un objeto de la clase Turtle y un largo. Dentro de la función se define el caso base dentro de un condicional que dibuja una línea mientras sea menor que el largo mínimo que es definido por el programador, y se llama así misma mientras el largo sea mayor al largo mínimo.

II. Análisis del Problema

II.1. Problema Propuesto

El problema propuesto tiene como desafío generar el copo de nieve de Koch, a través de la Curva de Koch utilizando el siguiente procedimiento:

- Se toma un segmento de largo L , el cual se divide en tres partes iguales de largo $L/3$.
- El segmento medio de esta división es reemplazado por 2 segmentos de lados $L/3$ que forma un ángulo de 60° grados (sexagesimales) entre ellos.
- Para los nuevos segmentos generados se repiten los pasos anteriores en las iteraciones posteriores.

En la Figura 1, se aprecia el procedimiento descrito y en la Figura 2, el Fractal o Copo de nieve de Koch.

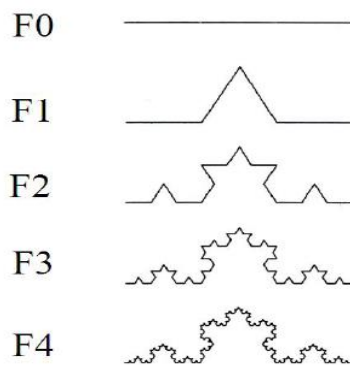


Figura 2 Iteraciones Curva de Koch

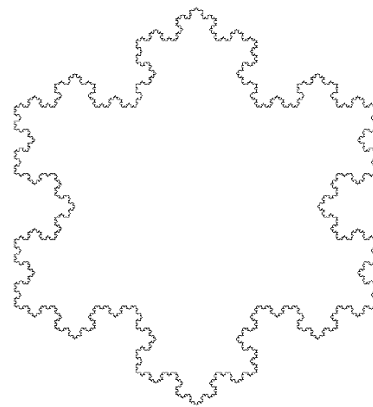


Figura 1 Copo de Nieve de Koch

Finalmente, el Copo de Nieve de Koch se traza siguiendo la figura de un triángulo equilátero, usando las Curvas de Koch como los lados del triángulo. Como se aprecia en la figura 3.

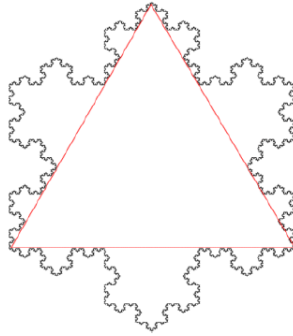


Figura 3 Trazo Copo de Koch, siguiendo un triángulo equilátero.

II.2. Implementación

Para la resolución del problema propuesto se utilizó el Lenguaje de Programación Orientado a Objetos, JAVA, desarrollado por James Gosling de Sun Microsystems en 1995 y el editor de código, Eclipse, desarrollado por IBM.

Para su implementación, se creó la clase pública *Koch* y dentro de ella se definieron las siguientes variables y funciones, que serán detalladas a continuación.

- Primero se definieron las variables estáticas *int LMIN*, que tomaba un pequeño valor entero, *int step*, encargada de tomar el largo mayor que la *Clase Turtle* debía trazar, y la variable *double radius*, que designa la distancia desde el centro hacia el contorno de la figura.
- Luego, se creó la función recursiva *public static void curvaDeKoch(Turtle tortuga, int largo)* que recibe una instancia de *Turtle* y un largo, y según esto, dibuja la curva de Koch que utiliza como caso base el largo mínimo *LMIN*, es decir, si la curva a dibujar es de largo menor a *LMIN* entonces se puede aproximar al segmento trazado.
- Finalmente, usando las funciones *public static void main(String[] args)* y *curvaDeKoch*, se creó el programa encargado de dibujar el copo de Koch siguiendo el esquema explicado en la *figura 3*. Que además pide el ángulo de rotación con respecto al centro de la figura al usuario.

III. Solución del Problema

III.1. Desarrollo

Para el desarrollo del problema propuesto, y profundizando en la implementación del punto anterior, se explicará a continuación el código desarrollado en JAVA del archivo koch.java elaborado por el alumno, separando en tres puntos importantes.

➤ Definición de Variables: Como se mencionó anteriormente, se crearon tres variables estáticas numéricas y una variable private de la clase Scanner dentro de la Clase Koch, que son:

1. int LMIN: variable numérica tipo entero, que da el largo mínimo que tiene los segmentos del fractal, el cual una vez que el largo es menor al largo mínimo se entra al caso base de la recursión.
2. int step: variable numérica tipo entero, determina el largo máximo de los segmentos que traza dentro del Canvas los objetos de tipo Turtle mediante la función curvaDeKoch.
3. double radius: variable numérica tipo double, determina el largo que el objeto de la clase Turtle recorre desde el centro hasta el contorno de la figura, para que así la está rote sobre su propio eje.
4. Scanner p: variable tipo scanner, que define un objeto de clase Scanner, que será usado para llamar a los métodos de dicha clase.

En la figura 4, se observa la definición de dichas variables.

```
1 import java.awt.Color;
2
3 import java.util.Scanner;
4
5 public class Koch {
6
7     static int LMIN = 3;
8     private static Scanner p;
9     static int step = 600;
10    static double radius = 280.0;
```

Figura 4

- Función `curvaDeKoch(Turtle tortuga, int largo)`: Función recursiva encargada de trazar los segmentos de la curva de Koch, recibe una instancia tortuga de la Clase `Turtle` y un `largo` entero del tipo `int`, su implementación puede dividirse en dos grandes puntos.
1. Caso base: Para cortar el ciclo recursivo se llama los condicionales de JAVA `if-else`, para el caso base, tenemos la condición que si `largo` es menor o igual `LMIN`, se llama al método `goForward` de la clase `Turtle`, que dibuja el trazado dependiendo del `step` definido dentro del código.
 2. Caso recursivo: Si no se cumple la condición anterior, pasamos al caso recursivo que llama a la función dentro de sí misma, dividiendo el `largo` por 3 hasta pasar al caso base. Para trazar los triángulos presentes en los segmentos, usamos los métodos `turnRight` y `turnLeft` de la clase `Turtle`, que modifican la dirección del curso de acuerdo al ángulo que se ingresa como parámetro a estos métodos.

En la figura 5, se puede apreciar el código de la función.

```
34 public static void curvaDeKoch(Turtle tortuga, int largo){
35
36
37     if(largo<=LMIN){
38
39         tortuga.goForward(largo);
40
41     }
42
43     else{
44
45         curvaDeKoch(tortuga,largo/3);
46         tortuga.turnLeft(60);
47         curvaDeKoch(tortuga,largo/3);
48         tortuga.turnRight(120);
49         curvaDeKoch(tortuga,largo/3);
50         tortuga.turnLeft(60);
51         curvaDeKoch(tortuga,largo/3);
52
53     }
54 }
```

Figura 5

- Función static public void main(String[] args): La función main, ejecuta el programa pidiendo al usuario el ángulo de inclinación que rota la figura, y realiza la instrucciones que dibujan el fractal de Koch. Detallando los principales puntos de funcionamiento.
1. Ángulo de inclinación: utilizando los métodos de la clase Scanner y definiendo la variable p, como una nueva instancia de clase Scanner, a través del output *System.out.println(...)* se pide al usuario que ingrese un ángulo sexagesimal, luego, dicho valor es guardado en la variable int ángulo, mediante el método nextInt() de la clase Scanner, este valor se utilizado para la rotación de la figura.
 2. Diseño Copo de Nieve de Koch: para el diseño del fractal, se define un nuevo objeto Turtle en una variable t, con sus respectivos paramentos. Finalmente usando la función curvaDeKoch y los métodos turnLeft y TurnRight, ingresando un ángulos de 60 y 120 grados para dichos métodos. Se sigue el esquema del triángulo.
 3. Bloque Try-Catch: Para indicar al usuario que no ingreso un valor numérico tipo int se despliega un mensaje de error, que indica que el valor ingresado no es correcto. La idea de esto era implementar un bucle que pidiera valor correcto hasta que el usuario lo ingresará, sin embargo, el alumno no pudo implementar este último punto.

En la figura 5, es posible ver el código de la función main.


```
12 static public void main(String[] args) {  
13  
14     p = new Scanner(System.in);  
15     int angulo;  
16  
17     System.out.print("Ingrese ángulo de inclinación: \n");  
18  
19     try {  
20  
21         angulo = p.nextInt();  
22  
23         Turtle t = new Turtle(600,350,90,1200,700);  
24         t.setPenColor(Color.GREEN);  
25         t.setPenUp();  
26         t.turnLeft(angulo);  
27         t.goForward(radius);  
28         t.turnRight(150);  
29         t.setPenDown();  
30         curvaDeKoch(t, step);  
31         t.turnRight(120);  
32         curvaDeKoch(t, step);  
33         t.turnRight(120);  
34         curvaDeKoch(t, step);  
35  
36     }  
37  
38     catch(Exception e){  
39  
40         System.out.println("No me diste un valor entero");  
41     }  
42  
43 }
```

Figura 5

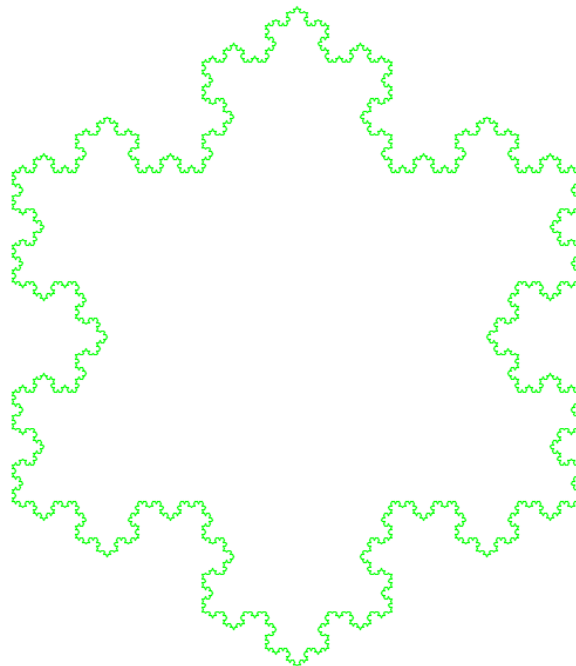
IV. Ejemplos de Funcionamiento

A continuación se dan algunos ejemplos de funcionamiento del código en ejecución.

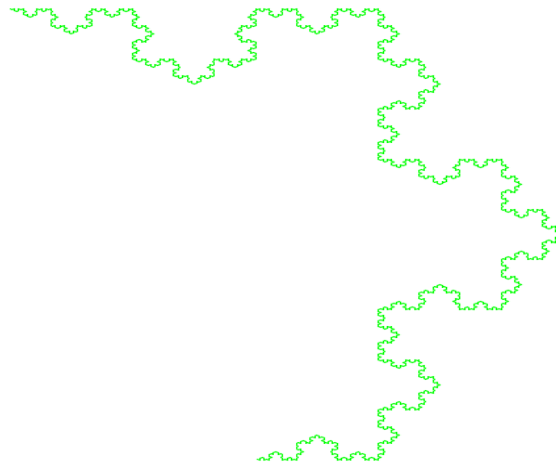
- El valor del ángulo en ingresado por el mediante, en la ventana de la consola de Eclipse.

```
Koch [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (27-03-2016 10:44:42 p. m.)  
Ingrese ángulo de inclinación: 45|
```

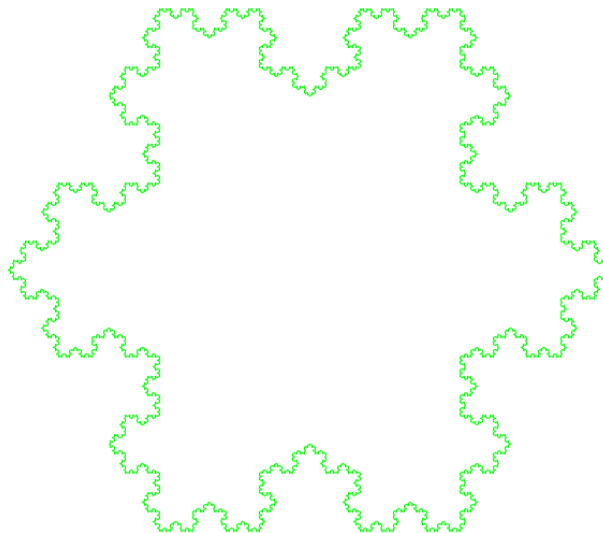
- Ejemplo de ejecución con ángulo de 0 grados.



- Ejecución del programa mientras se dibuja el fractal con ángulo de inclinación de 45 grados.



- Ejemplo del Copo de Nieve con ángulo de inclinación de 45 grados.



V. Modos de Uso

Para la compilación del fractal fue necesario utilizar los archivos stdDraw.java y Turtle.java que guardan las clases stdDraw y Turtle, respectivamente. Se dará una breve explicación de ella a continuación.

- Clase StdDraw: Esta clase facilita el uso de Canvas y el trabajo con dibujo de figuras geométricas en java usando Swing.
- Clase Turtle: Esta clase permite el dibujo sobre un Canvas usando el sistema Tortuga. Este sistema se basa en indicarle a la tortuga (clase) los movimientos y giros que se quieren realizar, a medida que la tortuga avanza una línea se traza por la trayectoria descrita. Posee un constructor que recibe 4 parámetros x0,y0,a0,w,h; que sitúa la tortuga en las coordenadas (x,y) con una orientación angular a0 en un Canvas de tamaño (w,h). Posee los métodos goForward, turnLeft y turnRight para avanzar en línea recta, cambiar la orientación angular a la izquierda y la derecha respectivamente.

VI. Resultados y Análisis

Dentro de los principales resultados, se logró implementar el Copo de Nieve de Koch utilizando funciones recursivas, que permitieron expresar una solución simple a un problema de múltiples iteraciones. Una de las principales complicaciones fue centrar el ángulo de rotación en el centro de la figura, pero mediante los métodos `setPenUp` y `setPenDown` se pudo hacer. En cuanto al uso de JAVA, se debe tener cuidado al definir los distintos tipos de variables e instancias de cada objeto, ya que estas ayudan a ver en donde pueden estar los errores en el código.

Como primera tarea, fue un buen ejercicio para familiarizarse con el manejo de clases, como llamar sus respectivos métodos, funciones y variables que a veces no son intuitivas, y que ayudaron a generar la programación necesaria de acuerdo a lo pedido en la Tarea y lo entendido por el alumno.