

# **Informe Tarea 1 Algoritmos y Estructuras de Datos.**

- **Número Tarea: 1**
- **Fecha: 30/03/2016**
- **Autor: Marcial Díaz Flores**
- **Email: [marcial.diaz@ing.uchile.cl](mailto:marcial.diaz@ing.uchile.cl)**
- **Código del curso: CC3001-02**

# **Introducción**

**La tarea consiste en implementar mediante Java un programa que pida al usuario un ángulo, y luego dibuje el copo de Koch rotado en dicho ángulo (Con respecto a la horizontal y al centro).**

**Esto se logra creando un código que usando Turtle (Clase para dibujar en Java) dibuje tres segmentos del copo de Koch y rote el dibujo según el ángulo ingresado por el usuario.**

# **Análisis del Problema**

**Entrando en los detalles del problema se tiene que dibujar el copo de Koch con un largo mínimo, si la curva a dibujar es más pequeña que dicha medida se aproxima como un segmento.**

**Una vez dibujado el primer segmento con un largo óptimo (de manera que el dibujo sea visualizado de la mejor manera posible) los 2 segmentos restantes del copo, y se ajustan los parámetros iniciales de turtle para que el dibujo esté centrado.**

**Al momento de resolver el problema se supuso que el usuario ingresará el ángulo medido en grados sexagesimales, dicho ángulo puede ser positivo o negativo, y mayor que  $360^\circ$ . También se supuso que el copo rotará desde el centro en un ángulo alfa formado con la horizontal, y el ángulo crece en sentido antihorario.**

**El problema se solucionó creando 3 funciones: segmentoKoch (recibe una instancia de turtle y un largo, luego dibuja un segmento de la curva de Koch según dicho largo), curvaDeKoch (dibuja el copo de Koch usando la función segmentoKoch 3 veces) y main (que pide el ángulo de rotación al**

usuario e inicializa una instancia de turtle en el centro, luego turtle se mueve a la posición donde empieza a dibujar el copo).

## Solución del Problema

Para dibujar un segmento del copo primero se fija un largo mínimo, en este caso 5.

Luego se utilizó la función `segmentoDeKoch`, que recibe una instancia de turtle y un largo: `segmentoDeKoch(Turtle tortuga, int largo)`.

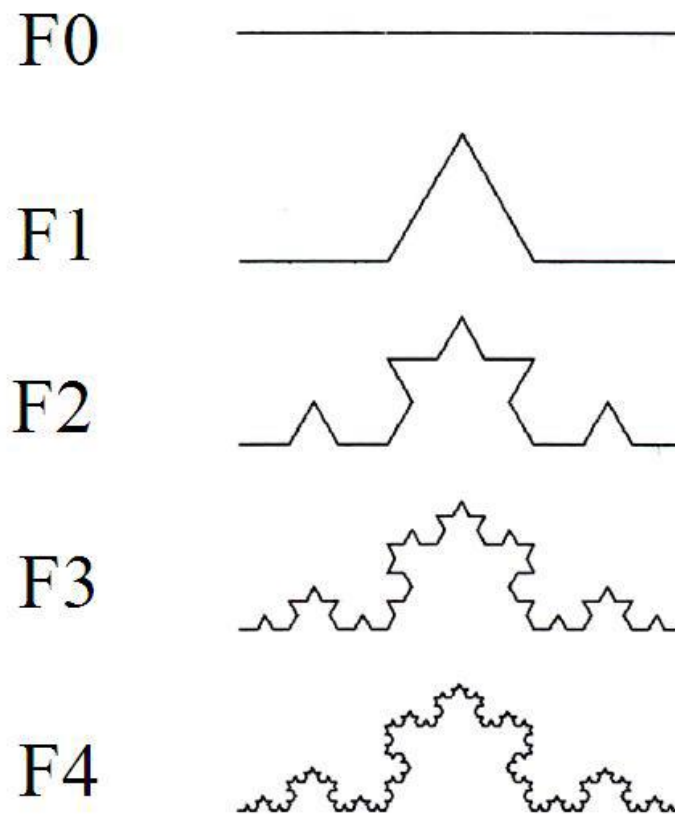
Dicha función utiliza el siguiente algoritmo:

**Caso base:** Si el largo ingresado es menor al largo mínimo la tortuga avanza hacia adelante en la medida “largo” (con la instrucción `tortuga.goForward(largo)`).

**Caso recursivo:** Si el largo es mayor al mínimo se dibujan cuatro segmentos de Koch de un largo igual a  $\text{largo}/3$ , usando la instrucción `segmentoDeKoch(tortuga, largo/3)`.

Cuando la tortuga acaba de dibujar el primero de estos cuatro segmentos gira  $60^\circ$  hacia la izquierda, luego dibuja el segundo y gira  $240^\circ$  hacia la izquierda, después dibuja el tercero y gira

**60° hacia la izquierda, para terminar al dibujar el cuarto segmento.**



*Figura 1: Primeras iteraciones del dibujo*

### **Código de la función segmentoDeKoch:**

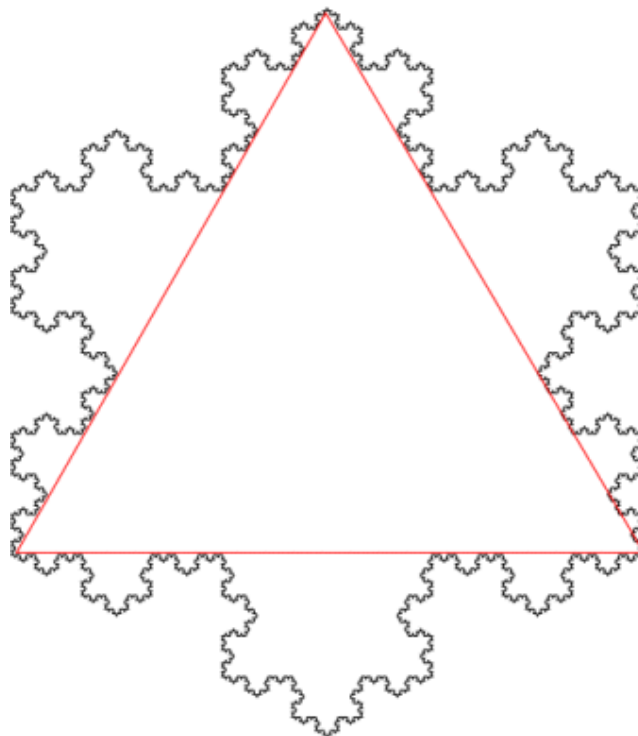
```
public static void segmentoKoch(Turtle tortuga, int largo) {  
    if (largo < LMIN)  
    {  
        tortuga.goForward(largo);  
    }  
    else  
    {  
        segmentoKoch(tortuga, largo/3);  
        tortuga.turnLeft(60);  
        segmentoKoch(tortuga, largo/3);  
        tortuga.turnLeft(240);  
        segmentoKoch(tortuga, largo/3);  
        tortuga.turnLeft(60);  
        segmentoKoch(tortuga, largo/3);  
    }  
}
```

**Después se implementó la función `curvaDeKoch`, que recibe una instancia de `turtle` y un `largo`: `curvaDeKoch(Turtle tortuga, int largo)`.**

**Dicha función usa el siguiente algoritmo:**

**Se dibuja hacia adelante un segmento del copo de Koch de medida “largo” (usando la función `segmentoDeKoch`), luego la tortuga gira  $240^\circ$  hacia la izquierda. Después se dibuja el segundo segmento, la tortuga vuelve a girar  $240^\circ$  hacia la izquierda y se dibuja el siguiente segmento.**

**Los tres segmentos forman un triángulo equilátero de lados “largo”, como se ve en la siguiente figura:**



*Figura 2: Copo de Koch sobre un triángulo equilátero*

## Código de la función curvaDeKoch:

```
public static void curvaDeKoch(Turtle tortuga, int largo) {  
    segmentoKoch(tortuga, largo);  
    tortuga.turnLeft(240);  
    segmentoKoch(tortuga, largo);  
    tortuga.turnLeft(240);  
    segmentoKoch(tortuga, largo);  
}
```

**La función main se encarga de solucionar el problema ejecutando las funciones anteriores con valores que fueron pensados para optimizar el dibujo, todo de la siguiente manera:**

**Se fija el largo de dibujo en 650, luego se le pregunta al usuario el ángulo de rotación del copo, inicializando una variable tipo Scanner para leer la respuesta del usuario.**

**Luego se inicializa una instancia de Turtle con los siguientes valores:**

- **Posición inicial:  $x_0 = 350$  e  $y_0 = 289$  (la tortuga parte en el centro)**
- **Ángulo inicial de la tortuga:  $a_0 = 180 + \alpha$  (con  $\alpha$  igual al ángulo ingresado por el usuario)**
- **Tamaño del canvas:  $w = 700$  y  $h = 578$**

**Después se levanta el lápiz de la tortuga (usando `tortuga.setPenUp()`) y la tortuga avanza 247 hacia**

**el frente, luego gira hacia la izquierda en  $270^\circ$ , avanza 143 hacia el frente y gira hacia la izquierda en  $270^\circ$ .**

**Hecho esto se baja el lápiz de la tortuga (usando `tortuga.setPenDown()`) y se dibuja la curva de Koch.**

**La idea de hacer que la tortuga parta en el centro mirando hacia la izquierda facilita la rotación del copo, ya que esta pasa a depender sólo del ángulo inicial de la tortuga (definido por  $180^\circ + \alpha$ ), luego se levanta el lápiz de la tortuga y esta se lleva a la posición inicial de dibujo (que también estará rotada).**

**Código de la función main:**

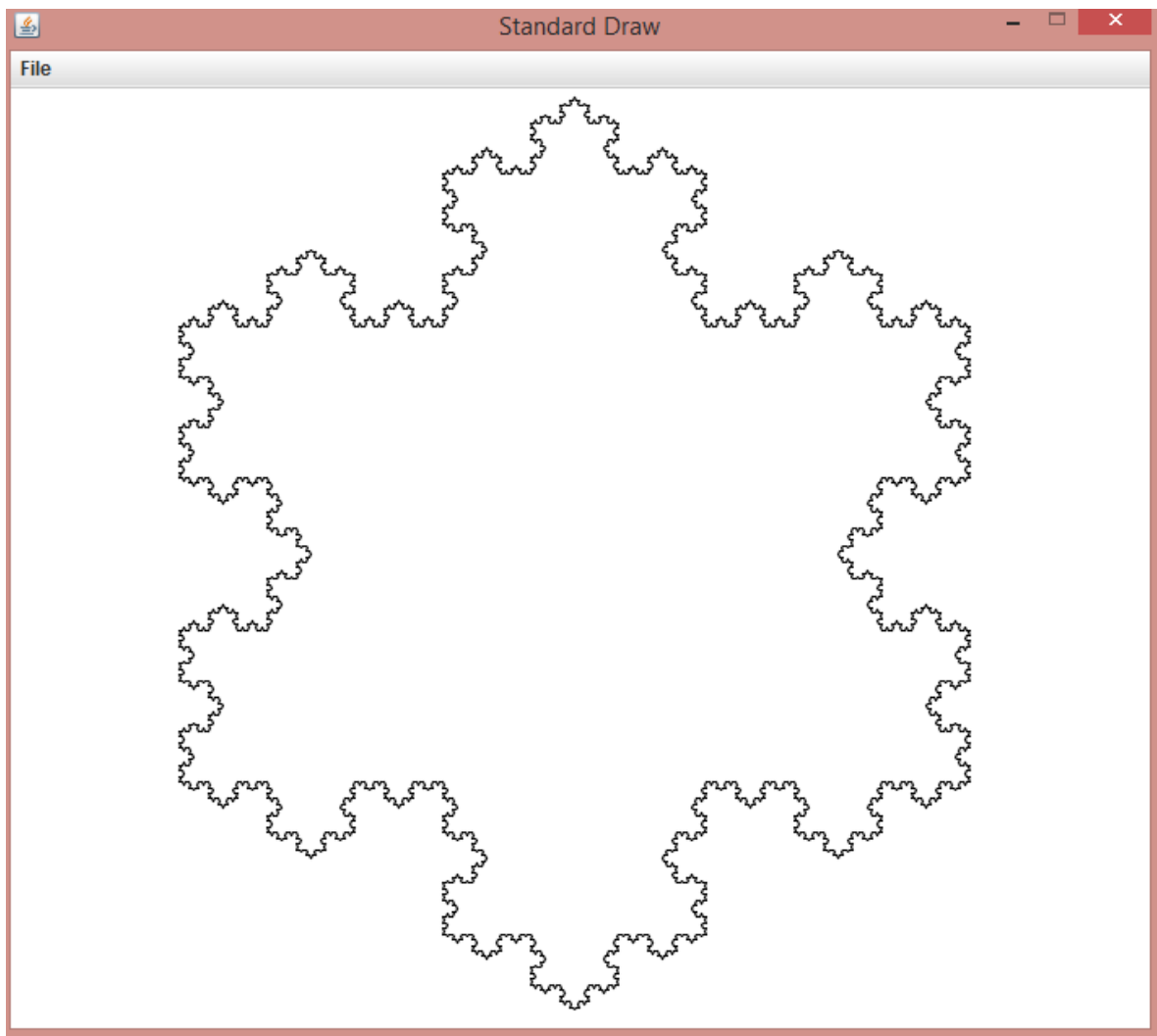
```
public static void main(String[ ] args){
    int largo = 650;
    System.out.println("Ingrese el ángulo alfa con el"
        + " que desea rotar el copo de Koch");
    System.out.println("El copo será rotado desde el"
        + " centro con un ángulo alfa respecto a"
        + " la horizontal, en sentido antihorario");
    Scanner lector = new Scanner(System.in);
    int alfa = lector.nextInt();
    Turtle tortuga;
    tortuga = new Turtle(350, 289, 180 + alfa, 700, 578);
    tortuga.setPenUp();
    tortuga.goForward(247);
    tortuga.turnLeft(270);
    tortuga.goForward(143);
    tortuga.turnLeft(270);
    tortuga.setPenDown();
    curvaDeKoch(tortuga, largo);
}
```



## Ejemplos de entradas y salidas:

```
Output - Tarea1 (run)
run:
Ingrese el ángulo alfa con el que desea rotar el copo de Koch
(El copo será rotado desde el centro con un ángulo alfa respecto a la horizontal, en sentido antihorario)
0
```

**Figura 3: Entrada =  $0^\circ$**

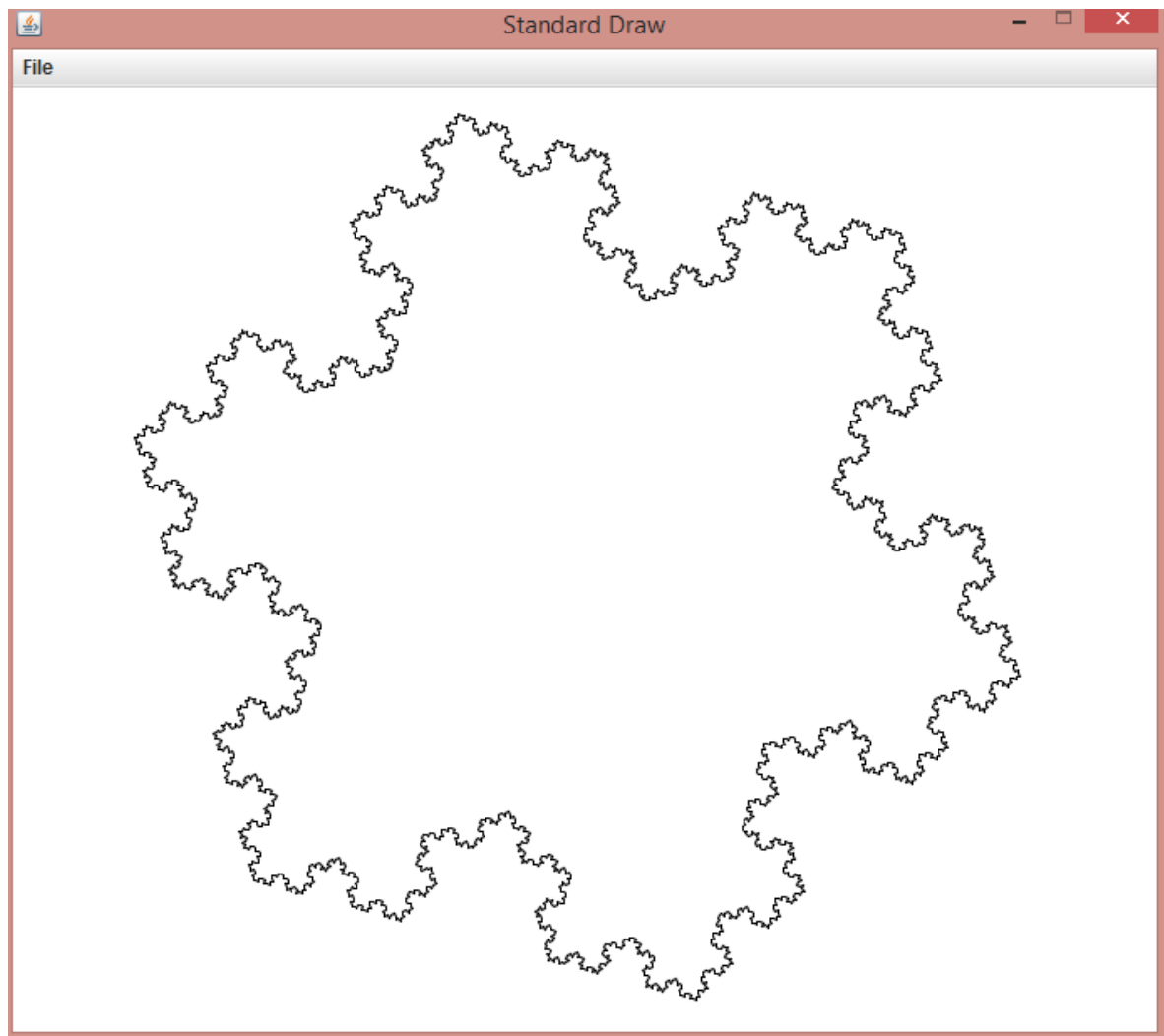


**Figura 4: Salida = El copo dibujado sin rotación**

```
Output - Tarea1 (run)

run:
Ingrese el ángulo alfa con el que desea rotar el copo de Koch
(El copo será rotado desde el centro con un ángulo alfa respecto a la horizontal, en sentido antihorario)
15
```

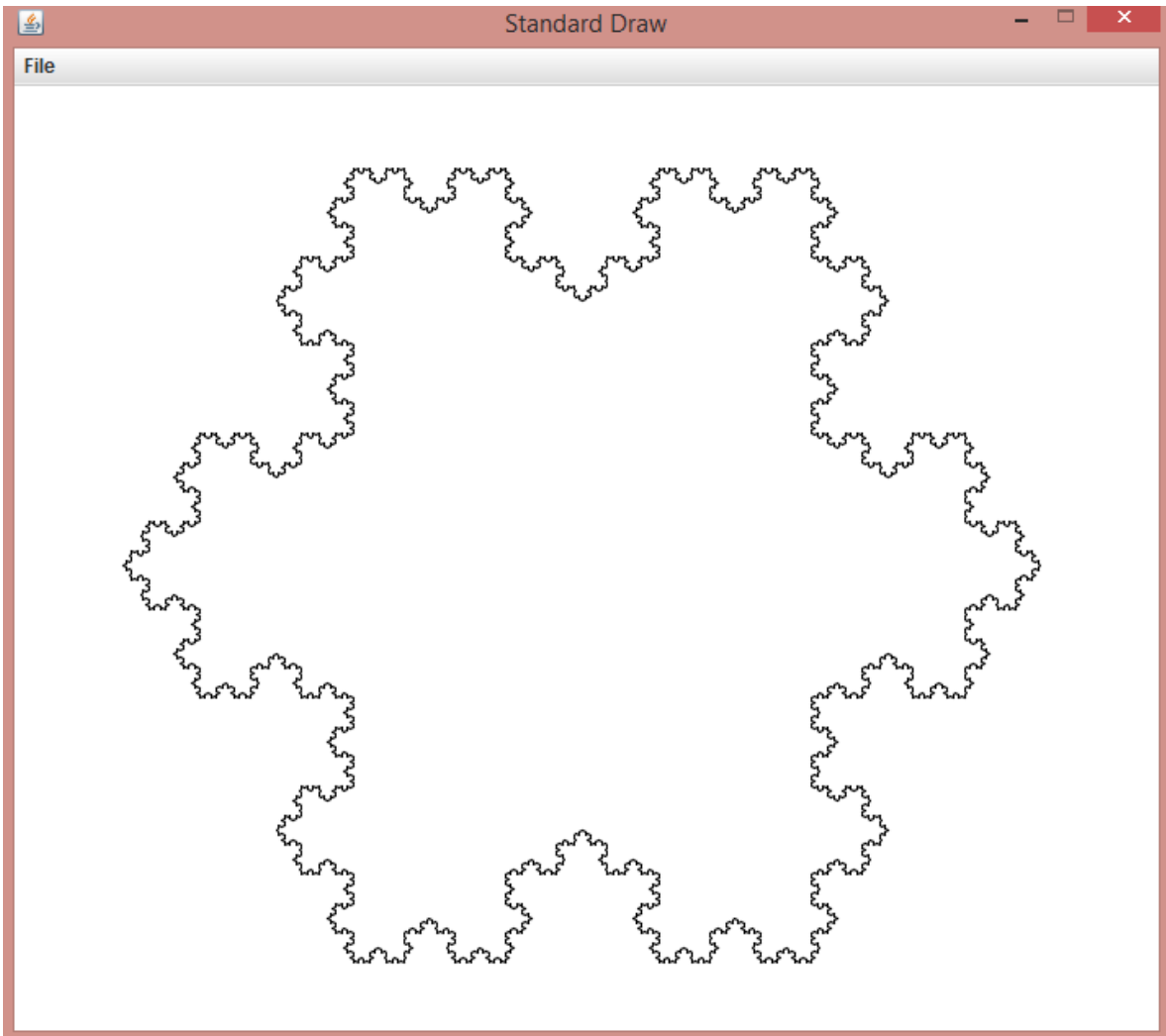
*Figura 5: Entrada = 15°*



*Figura 6: Salida = Copo rotado en 15° (sentido antihorario)*

```
Output - Tarea1 (run)
run:
Ingrese el ángulo alfa con el que desea rotar el copo de Koch
(El copo será rotado desde el centro con un ángulo alfa respecto a la horizontal, en sentido antihorario)
90
```

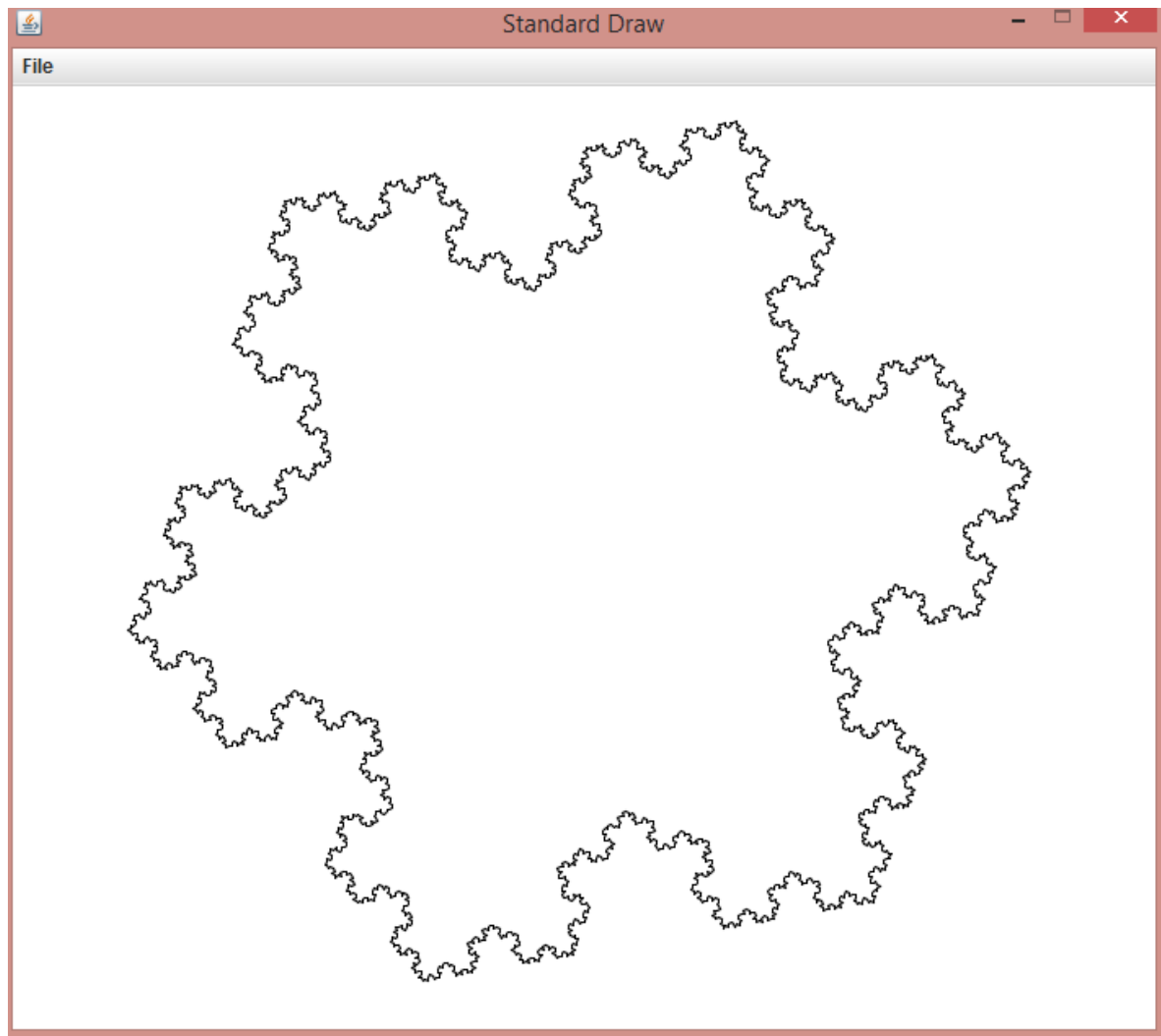
*Figura 7: Entrada = 90°*



*Figura 8: Salida = Copo rotado en 90° (sentido antihorario)*

```
Output - Tarea1 (run)
run:
Ingrese el ángulo alfa con el que desea rotar el copo de Koch
(El copo será rotado desde el centro con un ángulo alfa respecto a la horizontal, en sentido antihorario)
-380
```

*Figura 9: Entrada =  $-380^\circ$*



*Figura 10: Salida = Copo rotado  $380^\circ$  (en sentido horario)*

## **Modo de uso**

**Para ejecutar el programa se deben tener los archivos “Tarea1.java”, “Turtle.java” y “StdDraw.java” en la misma carpeta, compilar Tarea1.java y ejecutarlo. Luego se ingresa el ángulo (sólo el número) y se presiona enter.**

**Cabe destacar que la tarea se resolvió programando en el IDE Netbeans.**

## **Resultados y Análisis**

**De los resultados se concluye que los fractales son estructuras basadas en la recursividad, siendo buenos ejemplos de esta sobre todo en computación.**