**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе №5**

По дисциплине «СиАОД»

На тему:

«Рекурсия. Фракталы»

Выполнил: студент группы БВТ1801

Алмамма Башар

Руководитель:

Кутейников Иван Алексеевич

Москва 2020

**Цель работы:**

Используя технологию модульного программирования разработать программу генерацию фрактала, заданного заранее преподавателем, в соответствии с индивидуальным заданием. Разработать графический интерфейс пользователя.

**Задача (Вариант 1):**

Реализовать генерацию заданного типа фрактала с применением рекурсивных функций. Добавить возможность задания глубины фрактала. Оценить глубину рекурсии. Построить таблицу зависимости времени построения от глубины фрактала. Заданный фрактал: Кривая Коха.

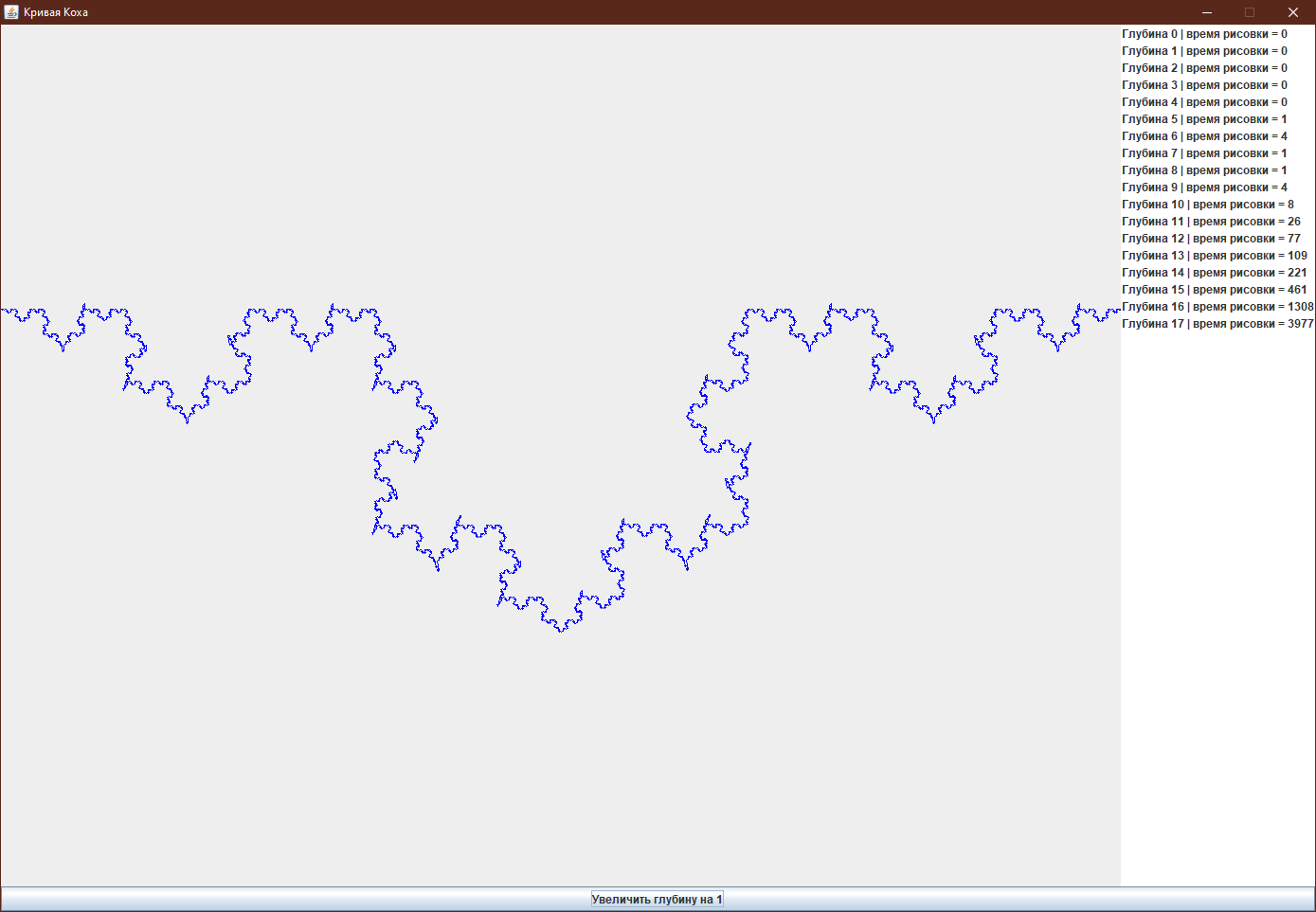
**Программа:**

**1.** Код программы:

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
  
public class KochSnowFlake extends javax.swing.JComponent {  
 public static int *width* = 1200, *height* = 900;  
 public static int *glubina* = 0;  
 private JList jList;  
 private DefaultListModel<String> list;  
 private long elapsedTime;  
  
 public static void main(String[] args) {  
 KochSnowFlake kochSnowFlake = new KochSnowFlake();  
 kochSnowFlake.setPreferredSize(new Dimension(*width*,*height*));  
 kochSnowFlake.creatAndShowGUI();  
 }  
  
 public void paint(Graphics g) {  
 g.setColor(Color.*blue*);  
 Point a = new Point(0, *height*/3);  
 Point b = new Point(this.getWidth(), *height*/3);  
  
 long start = System.*currentTimeMillis*();  
 drawFractal(g, a, b, 0, *glubina*);  
 long end = System.*currentTimeMillis*();  
 elapsedTime = end - start;  
 }  
  
 public void drawFractal(Graphics g, Point a, Point b, double d, int n) {  
 if (n <= 0) {  
 g.drawLine(a.x, a.y, b.x, b.y);  
 }  
 else {  
 double length = Math.*pow*(Math.*pow*(b.y - a.y, 2) + Math.*pow*(b.x - a.x, 2), 0.5);  
 double length1of3 = length / 3;  
 int round1 = (int) Math.*round*((length1of3 \* Math.*sin*(d)));  
 int round2 = (int) Math.*round*((length1of3 \* Math.*cos*(d)));  
 int round3 = (int) Math.*round*((length1of3 \* Math.*cos*(d + Math.*PI* / 3)));  
 int round4 = (int) Math.*round*((length1of3 \* Math.*sin*(d + Math.*PI* / 3)));  
  
 Point a1 = new Point(a.x + round2, a.y + round1); //1:3  
 Point b1 = new Point(a1.x + round2, a1.y + round1); //2:3  
 Point c = new Point(a1.x + round3, a1.y + round4); //вершина равн. треугольника  
  
 n--;  
 drawFractal(g, a1, c, d + Math.*PI* / 3, n);  
 drawFractal(g, c, b1, d - Math.*PI* / 3, n);  
  
 n--;  
 drawFractal(g, a, a1, d, n);  
 drawFractal(g, b1, b, d, n);  
 }  
 }

private void creatAndShowGUI(){  
 JFrame frame = new JFrame("Кривая Коха");  
 JButton button = new JButton("Увеличить глубину на 1");  
  
 list = new DefaultListModel<String>();  
 list.addElement("Глубина "+ *glubina* +" | время рисовки = " + elapsedTime );  
 jList = new JList(list);  
 jList.setSelectionMode(ListSelectionModel.*SINGLE\_INTERVAL\_SELECTION*);  
 jList.setLayoutOrientation(JList.*HORIZONTAL\_WRAP*);  
 jList.setVisibleRowCount(-1);  
 JScrollPane listScroller = new JScrollPane(jList);  
 listScroller.setPreferredSize(new Dimension(250, 80));  
  
 IncreaseEvent increaseEvent = new IncreaseEvent();  
 button.addActionListener(increaseEvent);  
  
 frame.add(jList,BorderLayout.*EAST*);  
 frame.add(this,BorderLayout.*CENTER*);  
 frame.add(button,BorderLayout.*SOUTH*);  
 frame.pack();  
 frame.setVisible(true);  
 frame.setResizable(false);  
 }  
 private void reDrawFrac(){  
 this.repaint();  
 list.addElement("Глубина "+ *glubina* +" | время рисовки = " + elapsedTime );  
 jList.setModel(list);  
 }  
  
 private class IncreaseEvent implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 *glubina*++;  
 reDrawFrac();  
 }  
 }  
}

**2.** Тестирование программы



**Вывод:** в данной лабораторной работе я научился реализовывать генерацию фракталов с помощью рекурсивных функций, а также создать графический интерфейс пользователя.