**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**

по предмету «СИАОД»

на тему:

«Рекурсия. Фракталы»

Выполнил: студент группы

Клюшкин Дмитрий Алексеевич

Руководитель:

Кутейников Иван Александрович

Москва 2020

*Цель работы:* 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.

*Вариант:* 14. Кривая Гильберта

*Ход работы:*

*Код программы:*

import javax.swing.\*;  
import java.awt.Graphics;  
  
public class Gilbert extends JFrame {  
 static SimpleGraphics *graphics*;

static int *p*; // порядок  
 static int *u*; // длина штриха  
  
 public Gilbert(int size){  
 setTitle("Gilbert");  
 setSize(size,size);  
 setVisible(true);  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 }  
  
 public static void main(String[] args) {  
 Gilbert gilbert = new Gilbert(900);  
 *graphics* = new SimpleGraphics(gilbert.getGraphics());  
 *p*=5; //порядок  
 *u*=(int)(850/Math.*pow*(2,*p*));  
 if (*u*==0) *u*=2;  
 *d*(*p*);  
 }  
  
 public void paint(Graphics g){}  
  
 static void a(int i)  
 {  
 if (i > 0)  
 {  
 *d*(i-1);  
 *graphics*.lineRel(*u*,0);  
 *a*(i-1);  
 *graphics*.lineRel(0, *u*);  
 *a*(i - 1);  
 *graphics*.lineRel(-*u*, 0);  
 *c*(i - 1);  
 }  
 }  
  
 static void b(int i)  
 {  
 if (i > 0)  
 {  
 *c*(i - 1);  
 *graphics*.lineRel(-*u*, 0);  
 *b*(i - 1);  
 *graphics*.lineRel(0, -*u*);  
 *b*(i - 1);  
 *graphics*.lineRel(*u*, 0);  
 *d*(i - 1);  
 }  
 }  
  
 static void c(int i)  
 {  
 if (i > 0)  
 {  
 *b*(i - 1);  
 *graphics*.lineRel(0, -*u*);  
 *c*(i - 1);  
 *graphics*.lineRel(-*u*, 0);  
 *c*(i - 1);  
 *graphics*.lineRel(0, *u*);  
 *a*(i - 1);  
 }  
 }  
 static void d(int i)  
 {  
 if (i > 0)  
 {  
 *a*(i - 1);  
 *graphics*.lineRel(0, *u*);  
 *d*(i - 1);  
 *graphics*.lineRel(*u*, 0);  
 *d*(i - 1);  
 *graphics*.lineRel(0, -*u*);  
 *b*(i - 1);  
 }  
 }  
  
 static class SimpleGraphics {  
 private Graphics g = null;  
 private int x = 20, y = 35;  
  
 public SimpleGraphics(Graphics g) {  
 this.g = g;  
 }  
  
 public void lineRel(int deltaX, int deltaY) {  
 g.drawLine(x, y, x + deltaX, y + deltaY);  
 x += deltaX;  
 y += deltaY;  
 }  
 }  
}

*Результат программы:*

|  |  |
| --- | --- |
| *Кривая Гильберта глубины 3* | *Кривая Гильберта глубины 5* |
|  |  |

|  |  |
| --- | --- |
| *Кривая Гильберта глубины 6* | *Кривая Гильберта глубины 7* |
|  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Глубина*** | *2* | *3* | *4* | *5* | *6* | *7* | *8* |
| ***Время в сек.*** | *0,002* | *0,003* | *0,012* | *0,045* | *0,273* | *1,902* | *20,236* |

Вывод: при увеличении глубины, время экспоненциально увеличивается. Получили знания с рекурсивным вызовом методов.