**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 №6**

По дисциплине «Кроссплатформенные технологии программирования»

На тему:

«Многопоточный генератор фракталов»

Выполнил: студент группы БВТ1801

Задоркин Максим Александрович

Руководитель:

Мосева Марина Сергеевна

Москва 2020

**Цель работы:** изучить основы многопоточности в Java.

**Задача:**В данной лабораторной работе необходимо будет реализовать возможность рисования фрактала с несколькими фоновыми потоками. Два преимущества данного подхода: первое - пользовательский интерфейс не будет зависать в процессе рисования нового фрактала, а второе - если у вас компьютер многоядерный, то процесс рисования будет намного быстрее. Несмотря на то, что многопоточное программирование может быть очень сложным, процесс изменения приложения будет прост, благодаря встроенной поддержке Swing фоновых потоков.   
Код программы представлен в репозитории git:   
<https://github.com/MaximZar/JavaOOP/tree/master/Laba6>

**Код для обработки потоков:**

private class FractalWorker extends SwingWorker<Object, Object> {

private int y;

int[] rgb;

public FractalWorker(int yCoord) {

y = yCoord;

}

public Object doInBackground() {

rgb = new int[dimensionDisplay];

for (int x = 0; x < dimensionDisplay; x += 1){

double xCoord = FractalGenerator.getCoord(complexRange.x, complexRange.x + complexRange.width,

dimensionDisplay, x);

double yCoord = FractalGenerator.getCoord(complexRange.y, complexRange.y + complexRange.height,

dimensionDisplay, y);

int numIters = generator.numIterations(xCoord, yCoord);

if (numIters == -1) {

rgb[x] = 0;

} else {

float hue = 0.7f + (float) numIters / 200f;

int rgbColor = Color.HSBtoRGB(hue, 1f, 1f);

rgb[x] = rgbColor;

}

}

return null;

}

public void done() {

for (int i = 0; i < dimensionDisplay; i += 1) {

image.drawPixel(i, y, rgb[i]);

}

image.repaint(0, 0, y, dimensionDisplay, 1);

rowsRemaining--;

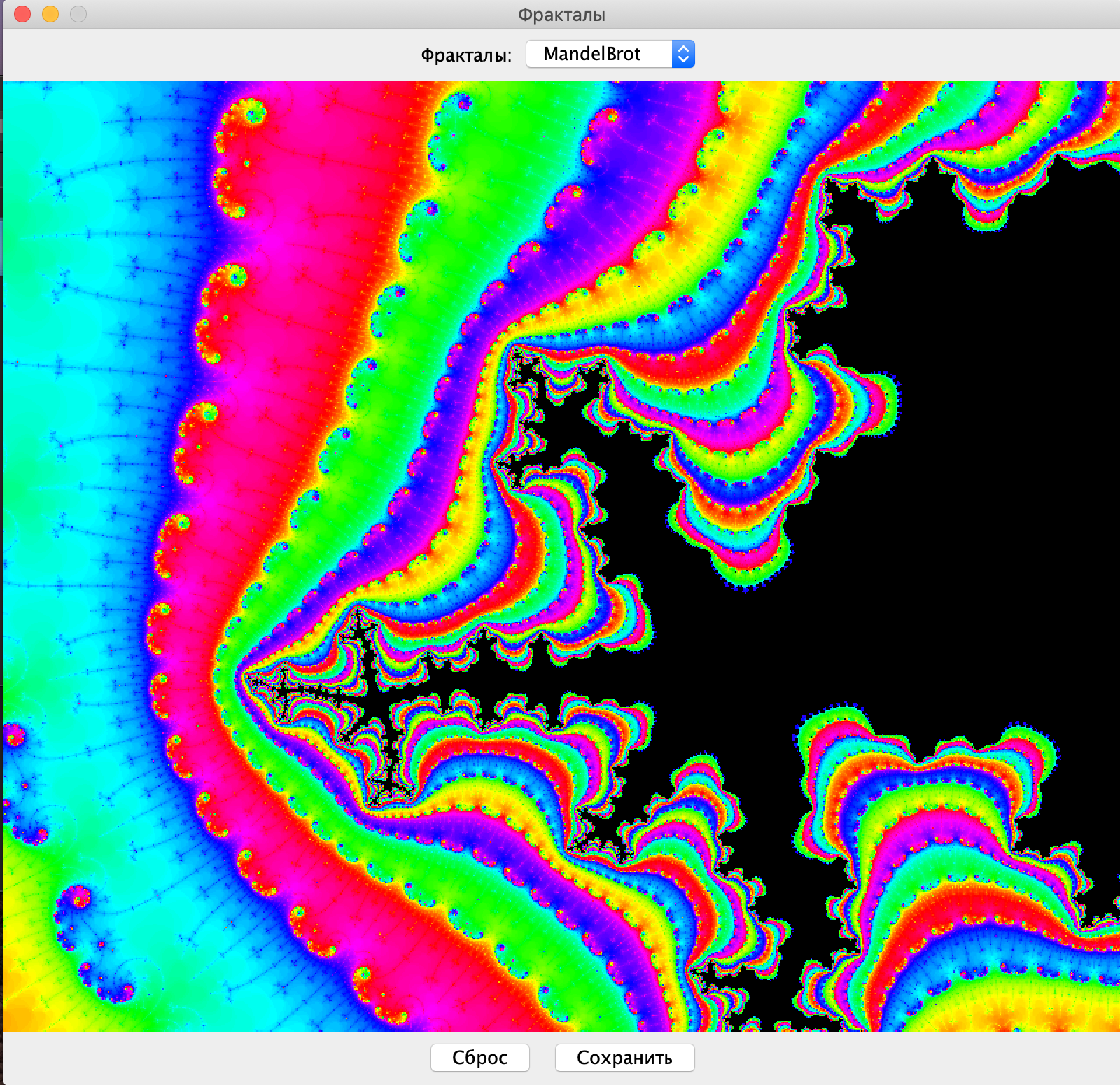
if (rowsRemaining == 0) {

enableUI(true);

}

}

}

**Работа программы:** 

В процесе отрисовки:

****

****

**Вывод:** в ходе лабораторной работы были изучены фракталы и базовая работа со swing. Рассмотрены базовые принципы построения многопоточности при помощи swing.