

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 «Технологии разработки программного обеспечения»

Выполнил: студент группы
БВТ1803

Маркушин Андрей Васильевич

Руководитель:

Мосева Марина Сергеевна

Москва 2020

Цель работы: создать небольшое JAVA-приложение, которое сможет рисовать фракталы.

Выполнение:

FractalExplorer:

```
import java.awt.*;
import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.event.*;
import java.awt.geom.Rectangle2D;
import java.awt.image.BufferedImage;
import java.io.File;

public class FractalExplorer {
    private int sizeDisplay;
    private JImageDisplay display;
    private FractalGenerator fractal;
    private Rectangle2D.Double range;

    static public void main(String args[]){
        FractalExplorer displayExplorer = new FractalExplorer(700);
        displayExplorer.createAndShowGUI();
        displayExplorer.drawFractal();
    }

    public FractalExplorer(int sizeDisplay) {
        this.sizeDisplay = sizeDisplay;
        range = new Rectangle2D.Double();
        fractal = new Mandelbrot();
        fractal.getInitialRange(range);
        display = new JImageDisplay(sizeDisplay, sizeDisplay);
    }

    public void createAndShowGUI() {
        display.setLayout(new BorderLayout());
        JFrame frame = new JFrame("Fractal");

        frame.add(display, BorderLayout.CENTER); //дисплей

        EventBtn eventBtn = new EventBtn(); //события кнопок
        EventMouse eventMouse = new EventMouse(); //события кнопок

        display.addMouseListener(eventMouse);

        JPanel DOWNpanel = new JPanel();
        JButton btnReset = new JButton("Reset");
        JButton btnSave = new JButton("Save Image");
        DOWNpanel.add(btnSave);
        DOWNpanel.add(btnReset);
        btnReset.addActionListener(eventBtn);
        btnSave.addActionListener(eventBtn);
        frame.add(DOWNpanel, BorderLayout.SOUTH); //кнопка сброса

        JPanel UPpanel = new JPanel(); //верхняя панель
        JComboBox comboBox = new JComboBox(); //комбо бокс
        comboBox.addActionListener(eventBtn);
        JLabel labelComboBox = new JLabel("Fractal:"); //подпись
        UPpanel.add(labelComboBox);
        UPpanel.add(comboBox);
        comboBox.addItem(new Mandelbrot());
    }
}
```

```

        comboBox.addItem(new Tricorn());
        comboBox.addItem(new BurningShip());
        frame.add(UPpanel, BorderLayout.NORTH); //добавить панель

        frame.pack (); //размещение
        frame.setVisible (true); //видимость
        frame.setResizable (false); //запрет изменения размера окна

        drawFractal();

        frame.repaint(); //обновить дисплей
    }

    private void drawFractal() {
        for (int x = 0; x < sizeDisplay; x++) {
            for (int y = 0; y < sizeDisplay; y++) {
                double xCoord = fractal.getCoord(range.x, range.x +
range.width, sizeDisplay, x);
                double yCoord = fractal.getCoord(range.y, range.y +
range.height, sizeDisplay, y);

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

                if (numIters == -1){
                    display.drawPixel(x, y, 0);
                }
                else {
                    float hue = 0.7f + (float) numIters / 200f;
                    int rgbColor = Color.HSBtoRGB(hue, 1f, 1f);
                    display.drawPixel(x, y, rgbColor);
                }
            }
        }
        display.repaint();
    }

    private class EventBtn implements ActionListener{
        public void actionPerformed(ActionEvent e)
        {
            if (e.getSource() instanceof JComboBox) {
                JComboBox comboBox = (JComboBox) e.getSource();
                fractal = (FractalGenerator) comboBox.getSelectedItem();
                fractal.getInitialRange(range);
                drawFractal();
            }
            else if (e.getActionCommand().equals("Reset")) {
                fractal.getInitialRange(range);
                drawFractal();
            }
            else if (e.getActionCommand().equals("Save Image")){
                JFileChooser openFileDialog = new JFileChooser();
                FileFilter filter = new FileNameExtensionFilter("PNG Images",
"png");

                openFileDialog.setFileFilter(filter);
                openFileDialog.setAcceptAllFileFilterUsed(false);
                int selectedFile = openFileDialog.showSaveDialog(display);
                if (selectedFile == openFileDialog.APPROVE_OPTION){
                    File file = openFileDialog.getSelectedFile();
                    try {
                        BufferedImage image = display.image;
                        ImageIO.write(image, "png", file);
                    }
                    catch (Exception exception) {
                        JOptionPane.showMessageDialog(display,
exception.getMessage(), "Cannot Save Image",
JOptionPane.ERROR_MESSAGE);
                    }
                }
            }
        }
    }

```

```

    }
}

private class EventMouse extends MouseAdapter {
    public void mouseClicked(MouseEvent e)
    {
        int x = e.getX();
        double xCoord = fractal.getCoord(range.x, range.x + range.width,
sizeDisplay, x);

        int y = e.getY();
        double yCoord = fractal.getCoord(range.y, range.y + range.height,
sizeDisplay, y);

        fractal.recenterAndZoomRange(range, xCoord, yCoord, 0.5);

        drawFractal();
    }
}
}

```

Вывод: реализовали возможность рисования фрактала с несколькими фоновыми потоками.