

**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 «Технологии разработки программного обеспечения»

Выполнил: студент группы  
БВТ1801

Клюшкин Дмитрий Алексеевич

Руководитель:

Мосева Марина Сергеевна

Москва 2019

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

**Выполнение:**

**Tricorn:**

```
import java.awt.geom.Rectangle2D;
public class Tricorn extends FractalGenerator{
    public static final int MAX_ITERATIONS = 2000;
    @Override
    public void getInitialRange(Rectangle2D.Double range) {
        range.x = -2;
        range.y = -2;
        range.width = 4;
        range.height = 4;
    }
    @Override
    public int numIterations(double x, double y) {
        Complex Z = new Complex(0,0);
        int i;
        for (i=0; i<MAX_ITERATIONS && Z.getPow2Abs() < 4; i++){
            Z.setSoprPow2();
            Z.addXY(x,y);
        }
        if (i == MAX_ITERATIONS) return -1;
        return i;
    }
    public String toString() {
        return "Tricorn";
    }
}
```

**BurningShip:**

```
import java.awt.geom.Rectangle2D;
public class BurningShip extends FractalGenerator{
    public static final int MAX_ITERATIONS = 2000;
    @Override
    public void getInitialRange(Rectangle2D.Double range) {
        range.x = -2;
        range.y = -2.5;
        range.width = 4;
        range.height = 3.5;
    }
    @Override
    public int numIterations(double x, double y) {
        Complex Z = new Complex(0,0);
        int i;
        for (i=0; i<MAX_ITERATIONS && Z.getPow2Abs() < 4; i++){
            Z.setAbsPow2();
            Z.addXY(x,y);
        }
        if (i == MAX_ITERATIONS) return -1;
        return i;
    }
    public String toString() {
        return "Burning Ship";
    }
}
```

## FractalExplorer:

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

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(600);
        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);
        frame.add(UPpanel, BorderLayout.NORTH); //добавить панель
        comboBox.addItem(new Mandelbrot());
        comboBox.addItem(new Tricorn());
        comboBox.addItem(new BurningShip());
        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);

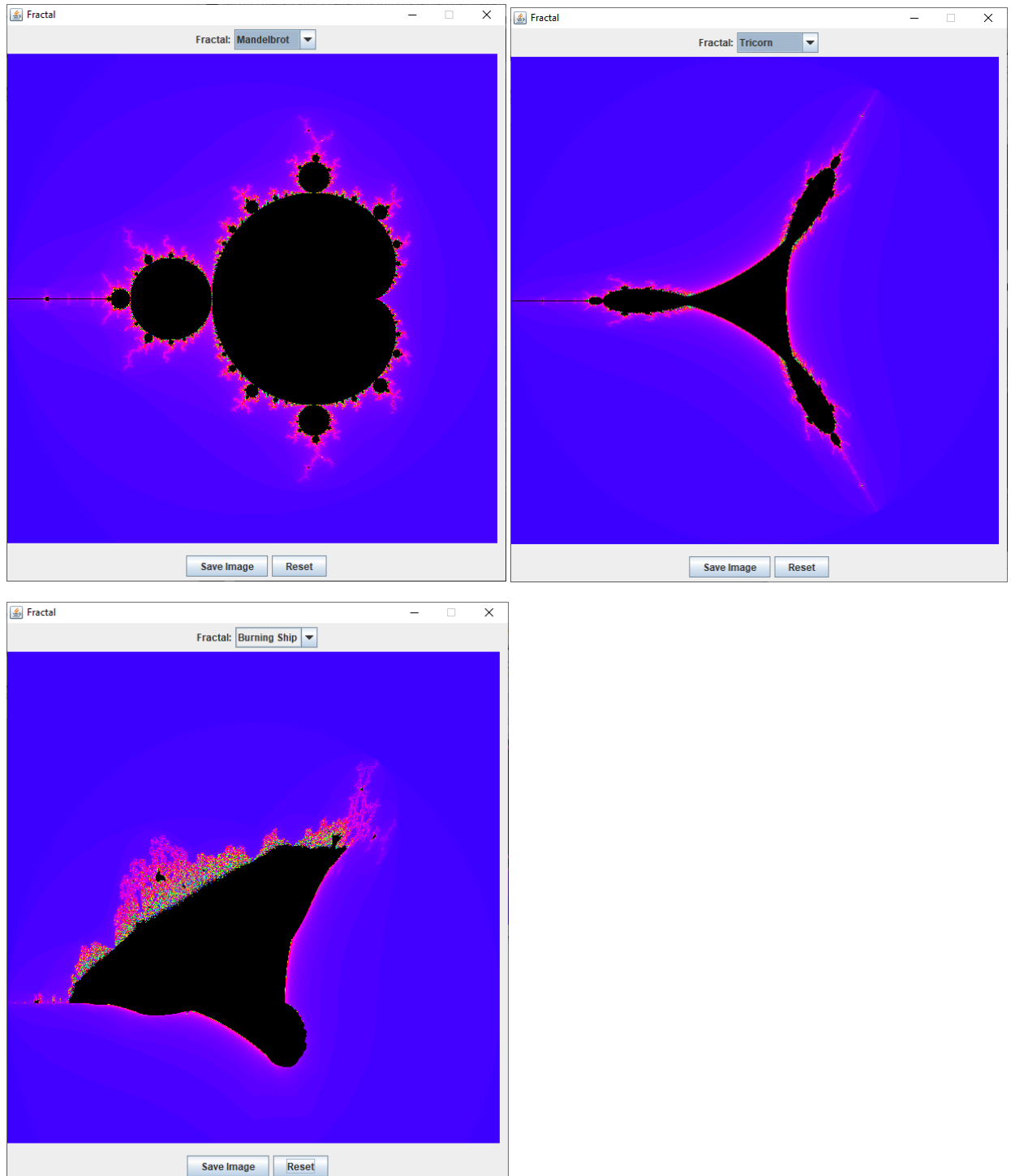
```

```

        int y = e.getY();
        double yCoord = fractal.getCoord(range.y, range.y + range.height,
sizeDisplay, y);
        fractal.recenterAndZoomRange(range, xCoord, yCoord, 0.5);
        drawFractal();
    }
}

```

## Скриншоты программы:



Вывод: добавил поддержку нескольких фракталов и реализовал возможность выбрать нужный фрактал из выпадающего списка, добавил поддержку сохранения текущего изображения в файл.