**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 БВТ1801

Клюшкин Дмитрий Алексеевич

Руководитель:

Мосева Марина Сергеевна

Москва 2019

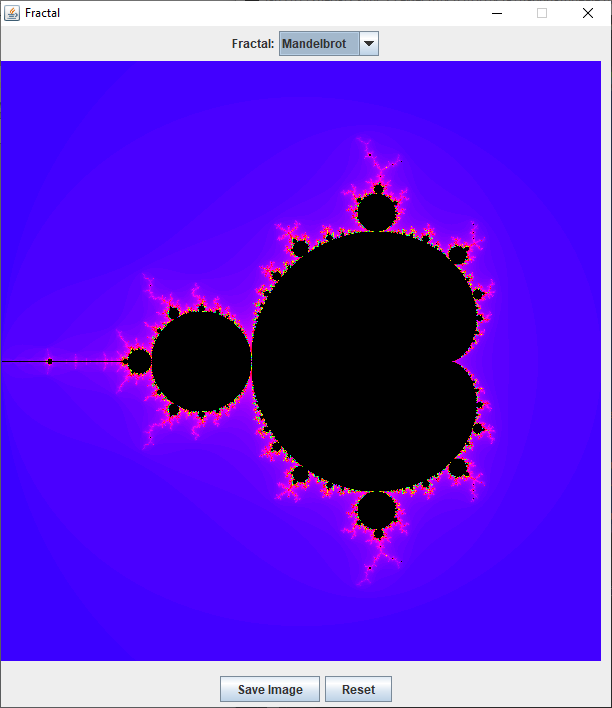
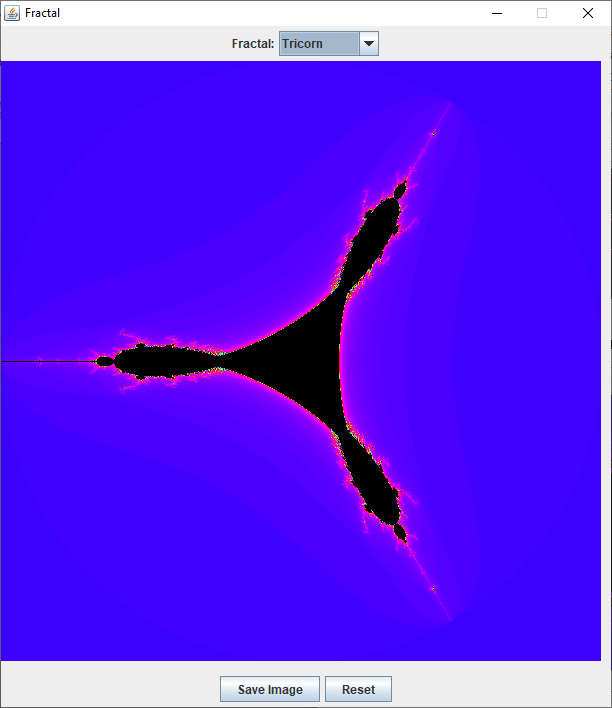
**Цель работы:** создать небольшое JAVA-приложение, которое сможет рисовать фракталы.

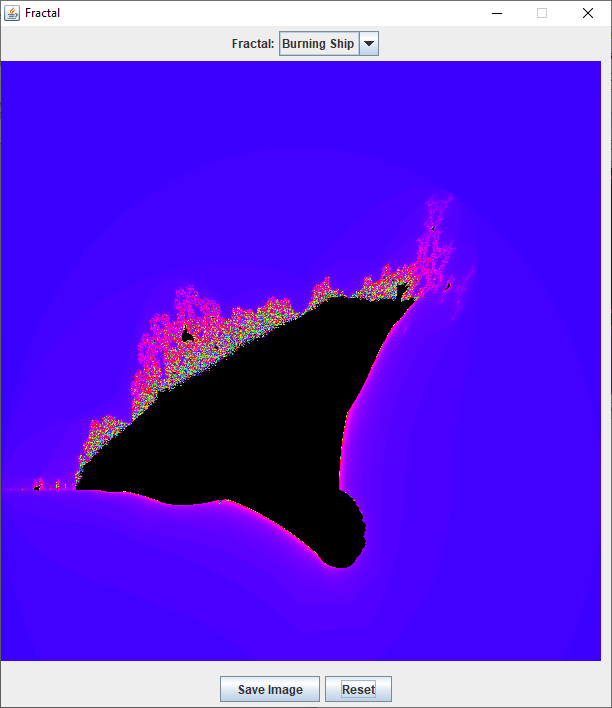
**Выполнение:**

**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;  
  
 JComboBox comboBox;  
 JButton btnReset;  
 JButton btnSave;  
 public int rowsRemaining=0;  
  
 static public void main(String args[]){  
 FractalExplorer displayExplorer = new FractalExplorer(600);  
 displayExplorer.createAndShowGUI();  
 }  
  
 public FractalExplorer(int size) {  
 sizeDisplay = size;  
 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();  
 btnReset = new JButton("Reset");  
 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  
 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  
  
 }  
  
 private void drawFractal() {  
 enableUI (false);  
  
 rowsRemaining = sizeDisplay;  
  
 for (int x=0; x<sizeDisplay; x++){  
 FractalWorker drawRow = new FractalWorker(x);  
 drawRow.execute();  
 }  
 }  
  
 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 openDialog = new JFileChooser();  
 FileFilter filter = new FileNameExtensionFilter("PNG Images", "png");  
 openDialog.setFileFilter(filter);  
 openDialog.setAcceptAllFileFilterUsed(false);  
 int selectedFile = openDialog.showSaveDialog(display);  
 if (selectedFile == openDialog.*APPROVE\_OPTION*){  
 File file = openDialog.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)  
 {  
 if (rowsRemaining != 0) {  
 return;  
 }  
  
 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();  
 }  
 }  
  
 private class FractalWorker extends SwingWorker<Object, Object> {  
 int yInit;  
 int[] rgbCoord;  
  
 private FractalWorker(int y) {  
 yInit = y;  
 }  
  
 protected Object doInBackground() {  
  
 rgbCoord = new int[sizeDisplay];  
  
 for (int x = 0; x < rgbCoord.length; x++) {  
 double xCoord = fractal.*getCoord*(range.x, range.x + range.width, sizeDisplay, x);  
 double yCoord = fractal.*getCoord*(range.y, range.y + range.height, sizeDisplay, yInit);  
  
 int numIters = fractal.numIterations(xCoord, yCoord);  
  
 if (numIters == -1) {  
 rgbCoord[x] = 0;  
 } else {  
 float hue = 0.7f + (float) numIters / 200f;  
 int rgbColor = Color.*HSBtoRGB*(hue, 1f, 1f);  
 rgbCoord[x] = rgbColor;  
 }  
 }  
 return null;  
  
 }  
  
 protected void done() {  
 for (int i = 0; i < rgbCoord.length; i++) {  
 display.drawPixel(i, yInit, rgbCoord[i]);  
 }  
 display.repaint(0, 0, yInit, sizeDisplay, 1);  
  
  
 rowsRemaining--;  
 if (rowsRemaining == 0) {  
 enableUI(true);  
 }  
 }  
  
 }  
  
 void enableUI(boolean activ){  
 comboBox.setEnabled(activ);  
 btnReset.setEnabled(activ);  
 btnSave.setEnabled(activ);  
 }  
}

**Скриншоты программы:**



Вывод: реализовали возможность рисования фрактала с несколькими фоновыми потоками.