МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

Московский технический университет связи и информатики

(МТУСИ)

Лабораторная работа №5 по теме

«Выбор и сохранение фрактала»

Выполнил:

Студент группы БФИ1801

Сиротин Н. С.

**Москва 2020**

**Цель работы:** В данной лабораторной работе генератор фракталов будет расширен двумя новыми функциями. Во-первых, вы добавите поддержку нескольких фракталов и реализуете возможность выбирать нужный фрактал из выпадающего списка. Во-вторых, вы добавите поддержку сохранения текущего изображения в файл. Ниже приведен скриншот, где продемонстрировано, как будет выглядеть новая программа

**Задачи:** Так как в реализацию была введена абстракция FractalGenerator, добавление нескольких фракталов не будет проблемой. В данной лабораторной работе вы добавите поддержку нескольких фракталов, и пользователь сможет выбирать между ними, используя combo-box. Программный интерфейс Swing (Swing API) предоставляет combo-box через класс javax.swing.JComboBox, а также запускает ActionEvents при выборе новогоэлемента.

Необходимо сделать:

• Создать 2 новые реализации FractalGenerator Первым будет фрактал tricorn, который должен находиться в файле Tricorn.java.

Для этого нужно создать подкласс FractalGenerator и реализация будет почти идентична фракталу Мандельброта, кроме двух изменений.

Следующая ваша задача - сохранение текущего изображения фрактала на диск.

Java API предоставляет несколько инструментов для реализации данной задачи.

• Во-первых, вам нужно добавить кнопку «Save Image» в ваше окно. Для этого вы можете добавить обе кнопки «Save Image» и «Reset» в новую Jpanel, а затем разместить эту панель в SOUTH части окна. События от кнопки «Save Image» также должны обрабатываться реализацией ActionListener. Назначьте кнопкам «Save Image» и «Reset» свои значения команд (например, «save» и «reset») для того, чтобы обработчик событий мог отличить события от этих двух разных кнопок.

• В обработчике кнопки «Save Image» вам необходимо реализовать возможность указания пользователем, в какой файл он будет сохранять изображение. Это можно сделать с помощью класса javax.swing.JFileChooser. Указанный класс предоставляет метод showSaveDialog(), который открывает диалоговое окно «Save file», позволяя тем самым пользователю выбрать директорию для сохранения.

Метод принимает графический компонент, который является родительским элементом для диалогового окна с выбором файла, что позволяет центрированию окна с выбором относительно его родителя. В качестве родителя используйте окно приложения. Как вы могли заметить, данный метод возвращает значение типа int, которое указывает результат операции выбора файла. Если метод возвращает значение JfileChooser.APPROVE\_OPTION, тогда можно продолжить операцию сохранения файлов, в противном случае, пользователь отменил операцию, поэтому закончите данную обработку события без сохранения. Если пользователь выбрал директорию для сохранения файла, вы можете ее узнать, используя метод getSelectedFile(), который возвращает объект типа File.

• Также необходимо настроить средство выбора файлов, чтобы сохранять изображения только в формате PNG, на данном этапе вы будете работать только с данным форматом. вы сможете это настроить с помощью javax.swing.filechooser.FileNameExtensionFilter, как это продемонстрировано ниже:

JFileChooser chooser = new JfileChooser();

FileFilter filter = new FileNameExtensionFilter("PNG Images", "png"); chooser.setFileFilter(filter);

chooser.setAcceptAllFileFilterUsed(false);

Последняя строка гарантирует, что средство выбора не разрешит пользователю использование отличных от png форматов.

• Если пользователь успешно выбрал файл, следующим шагом является сохранения изображения фрактала на диск!

**Код класса FractalExplorer:**

import java.awt.\*;  
import javax.imageio.ImageIO;  
import javax.swing.\*;  
import javax.swing.border.Border;  
import javax.swing.filechooser.FileNameExtensionFilter;  
import java.awt.event.\*;  
import java.awt.geom.Rectangle2D;  
import java.io.File;  
import java.io.FileFilter;  
import java.io.IOException;  
  
public class FractalExplorer {  
 private int displaySize;  
 private JImageDisplay imageDisplay;  
 private FractalGenerator fractalGenerator;  
 private Rectangle2D.Double range;  
 private JComboBox comboBox;  
  
 private FractalExplorer (int displaySize) {  
 this.displaySize = displaySize;  
 this.fractalGenerator = new Mandelbrot();  
 this.range = new Rectangle2D.Double(0,0,0,0);  
 fractalGenerator.getInitialRange(this.range);  
 }  
  
 // точка входа в программу  
 public static void main(String[] args) {  
 FractalExplorer fractalExplorer = new FractalExplorer(450);  
 fractalExplorer.setGUI();  
 fractalExplorer.drawFractal();  
 }  
  
 // здесь интерфейс  
 public void setGUI() {  
 JFrame frame = new JFrame("Разные фракталы 2021");  
 JButton buttonReset = new JButton("Очистить");  
 JButton buttonSave = new JButton("Сохранить изображение");  
 JPanel jPanel\_1 = new JPanel();  
 JPanel jPanel\_2 = new JPanel();  
 JLabel label = new JLabel("Фрактал:");  
  
 imageDisplay = new JImageDisplay(displaySize, displaySize);  
 imageDisplay.addMouseListener(new MouseListener());  
  
 // здесь список фракталов  
 comboBox = new JComboBox();  
 comboBox.addItem(new Mandelbrot());  
 comboBox.addItem(new Tricorn());  
 comboBox.addItem(new BurningShip());  
 comboBox.addActionListener(new ActionHandler());  
  
 // кнопка удаления фрактала  
 buttonReset.setActionCommand("Очистить");  
 buttonReset.addActionListener(new ActionHandler());  
  
 // кнопка сохранения фрактала  
 buttonSave.setActionCommand("Сохранить");  
 buttonSave.addActionListener(new ActionHandler());  
  
 jPanel\_1.add(label, BorderLayout.CENTER);  
 jPanel\_1.add(comboBox, BorderLayout.CENTER);  
 jPanel\_2.add(buttonReset, BorderLayout.CENTER);  
 jPanel\_2.add(buttonSave, BorderLayout.CENTER);  
  
 frame.setLayout(new BorderLayout());  
 frame.add(imageDisplay, BorderLayout.CENTER);  
 frame.add(jPanel\_1, BorderLayout.NORTH);  
 frame.add(jPanel\_2, BorderLayout.SOUTH);  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
  
 frame.pack();  
 frame.setVisible(true);  
 frame.setResizable(false);  
 }  
  
 // здесь картинка с фракталом  
 private void drawFractal() {  
 for (int x = 0; x < displaySize; x++) {  
 for (int y = 0; y < displaySize; y++) {  
 int counter = fractalGenerator.numIterations(FractalGenerator.getCoord(range.x, range.x + range.width, displaySize, x),  
 fractalGenerator.getCoord(range.y, range.y + range.width, displaySize, y));  
 if (counter == -1) {  
 imageDisplay.drawPixel(x, y, 0);  
 }  
 else {  
 float hue = 0.1f + (float) counter / 100f;  
 int rgbColor = Color.HSBtoRGB(hue, 1f, 1f);  
 imageDisplay.drawPixel(x, y, rgbColor);  
 }  
 }  
 }  
 imageDisplay.repaint();  
 }  
  
 // здесь обратотчик кнопок  
 public class ActionHandler implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getActionCommand().equals("Очистить")) {  
 // перерисовка фрактала  
 fractalGenerator.getInitialRange(range);  
 drawFractal();  
 } else if (e.getActionCommand().equals("Сохранить")) {  
 // сохранение  
 JFileChooser fileChooser = new JFileChooser();  
 FileNameExtensionFilter fileFilter = new FileNameExtensionFilter("ПНГ картинка", "png");  
 fileChooser.setFileFilter(fileFilter);  
 fileChooser.setAcceptAllFileFilterUsed(false);  
 int t = fileChooser.showSaveDialog(imageDisplay);  
 if (t == JFileChooser.APPROVE\_OPTION) {  
 try {  
 ImageIO.write(imageDisplay.getImage(), "png", fileChooser.getSelectedFile());  
 } catch (NullPointerException | IOException ee) {  
 JOptionPane.showMessageDialog(imageDisplay, ee.getMessage(), "Не удалось сохранить!", JOptionPane.ERROR\_MESSAGE);  
 }  
 }  
 } else {  
 fractalGenerator = (FractalGenerator) comboBox.getSelectedItem();  
 range = new Rectangle2D.Double(0,0,0,0);  
 fractalGenerator.getInitialRange(range);  
 drawFractal();  
 }  
 }  
 }  
  
  
 //обработчик собыйти клика мышкой  
 public class MouseListener extends MouseAdapter {  
 @Override  
 public void mouseClicked(MouseEvent e) {  
 double x = FractalGenerator.getCoord(range.x, range.x + range.width, displaySize, e.getX());  
 double y = FractalGenerator.getCoord(range.y, range.y + range.width, displaySize, e.getY());  
 fractalGenerator.recenterAndZoomRange(range, x, y, 0.5);  
 drawFractal();  
 }  
 }  
}

**Код класса FractalGenerator:**

import java.awt.geom.Rectangle2D;  
  
  
/\*\*  
 \* This class provides the common interface and operations for fractal  
 \* generators that can be viewed in the Fractal Explorer.  
 \*/  
public abstract class FractalGenerator {  
  
  
 public static double getCoord(double rangeMin, double rangeMax,  
 int size, int coord) {  
  
 assert size > 0;  
 assert coord >= 0 && coord < size;  
  
 double range = rangeMax - rangeMin;  
 return rangeMin + (range \* (double) coord / (double) size);  
 }  
  
  
 public abstract void getInitialRange(Rectangle2D.Double range);  
  
  
 public void recenterAndZoomRange(Rectangle2D.Double range,  
 double centerX, double centerY, double scale) {  
  
 double newWidth = range.width \* scale;  
 double newHeight = range.height \* scale;  
  
 range.x = centerX - newWidth / 2;  
 range.y = centerY - newHeight / 2;  
 range.width = newWidth;  
 range.height = newHeight;  
 }  
  
  
 public abstract int numIterations(double x, double y);  
}

**Код класса JImageDisplay:**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.image.BufferedImage;  
  
public class JImageDisplay extends JComponent {  
 private BufferedImage image;  
  
 // тут объявляется картинка  
 public JImageDisplay(int width, int height) {  
 image = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_RGB);  
 Dimension dimension = new Dimension(width, height);  
 super.setPreferredSize(dimension);  
 }  
  
 // тут отрисовывается фрактал  
 public void paintComponent(Graphics graphics) {  
 graphics.drawImage(image, 0, 0, image.getWidth(), image.getHeight(), null);  
 }  
  
 //тут удаляется фрактал  
 public void clearImage() {  
 for (int i = 0; i < image.getWidth(); i++) {  
 for (int j = 0; j < image.getHeight(); j++) {  
 drawPixel(i, j, 0);  
 }  
 }  
 }  
  
 // тут окрашивается каждый пиксель  
 public void drawPixel(int x, int y, int rgbColor) {  
 image.setRGB(x, y, rgbColor);  
 }  
  
 public BufferedImage getImage() {  
 return image;  
 }  
}

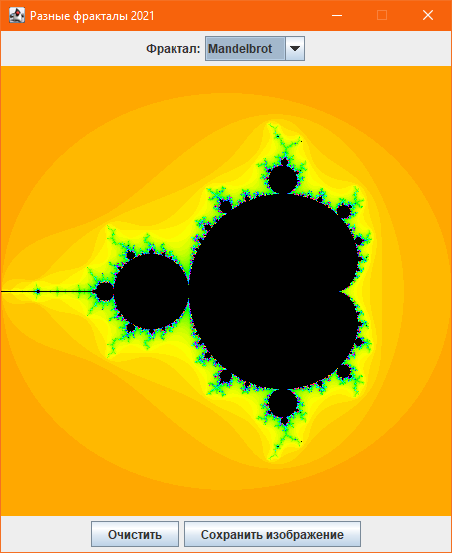


Рисунок 1 — Отображение фрактала Мандельброта

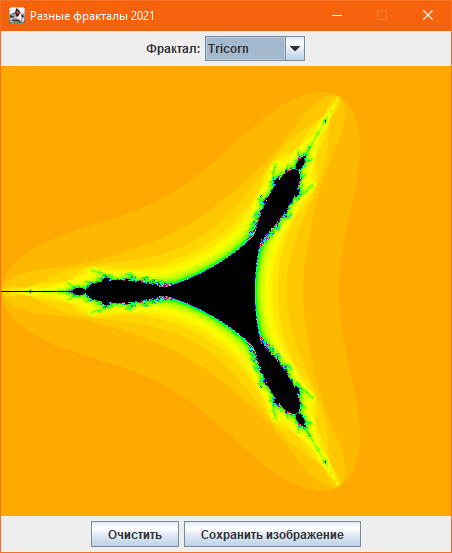


Рисунок 2 — Отображение фрактала Tricorn

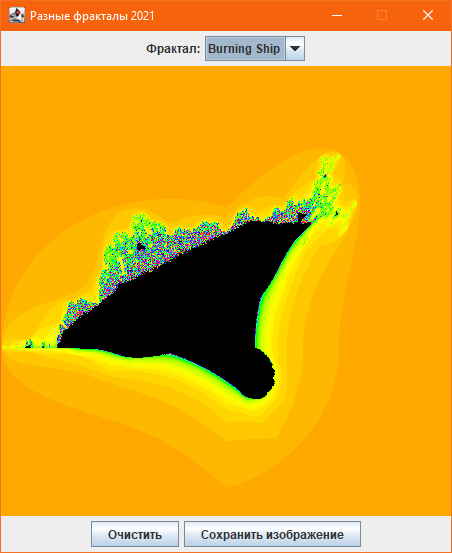


Рисунок 3 — Отображение фрактала Buring Ship

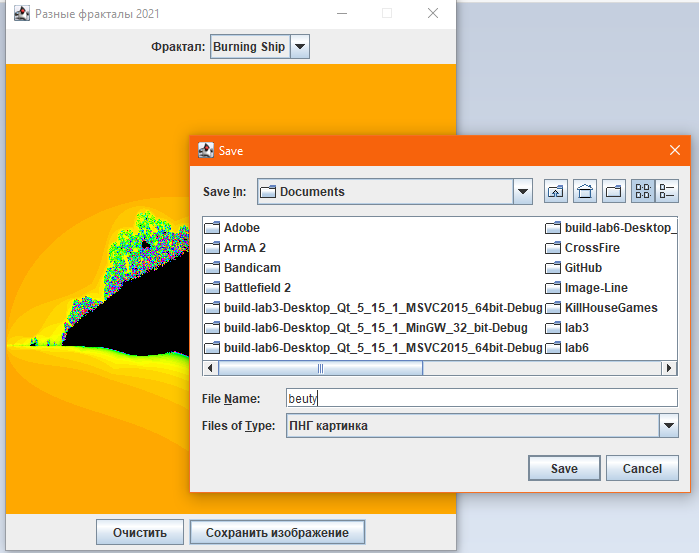


Рисунок 4 — Сохранение фрактала на компьютер

**Вывод:** Мы успешно разработали программу для генерации и отображения фракталов разных видов и их последующего сохранения на компьютере.