**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»**

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе № 5

по дисциплине «Информационные технологии и программирование»

Выполнил: студент группы БФИ2102

Голубева Елена Александровна

Проверила:

Мосева Марина Сергеевна

Москва, 2022

Цель работы

Добавление в JAVA-приложение поддержку нескольких фракталов и реализация возможности выбора нужного фрактала из выпадающего списка и поддержку сохранения текущего изображения в файл.

Задание

• Создать 2 новые реализации FractalGenerator

- Первым будет фрактал tricorn, который должен находиться в файле Tricorn.java.

- Второй фрактал, который необходимо реализовать - это фрактал «Burning Ship».

• Настроить JComboBox в вашем пользовательском интерфейсе можно с использованием конструктора без параметров, а затем использовать метод addItem(Object). Необходимо будет также добавить объект label в разрабатываемый пользовательский интерфейс перед выпадающим списком, в качестве пояснения к выпадающему списку. И наконец, необходимо добавить поддержку выпадающего списка в реализацию ActionListener.

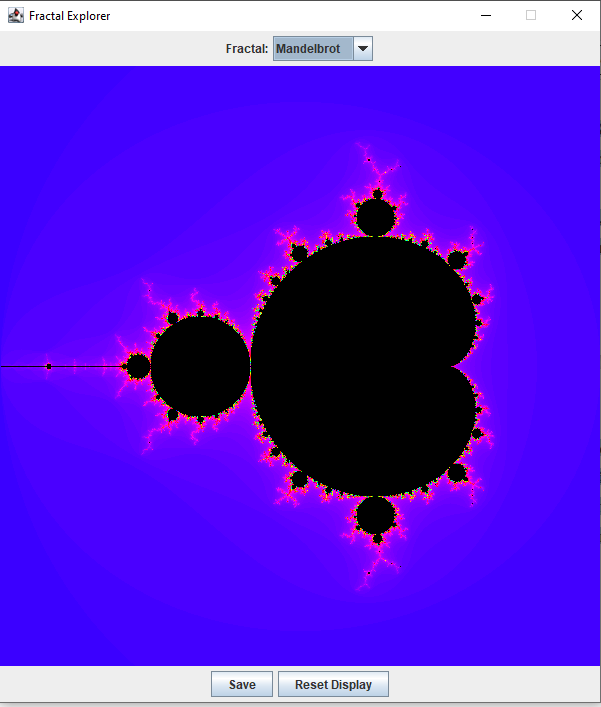
• Во-первых, вам нужно добавить кнопку «Save Image» в ваше окно. Для этого вы можете добавить обе кнопки «Save Image» и «Reset» в новую Jpanel, а затем разместить эту панель в SOUTH части окна. События от кнопки «Save Image» также должны обрабатываться реализацией ActionListener. Назначьте кнопкам «Save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

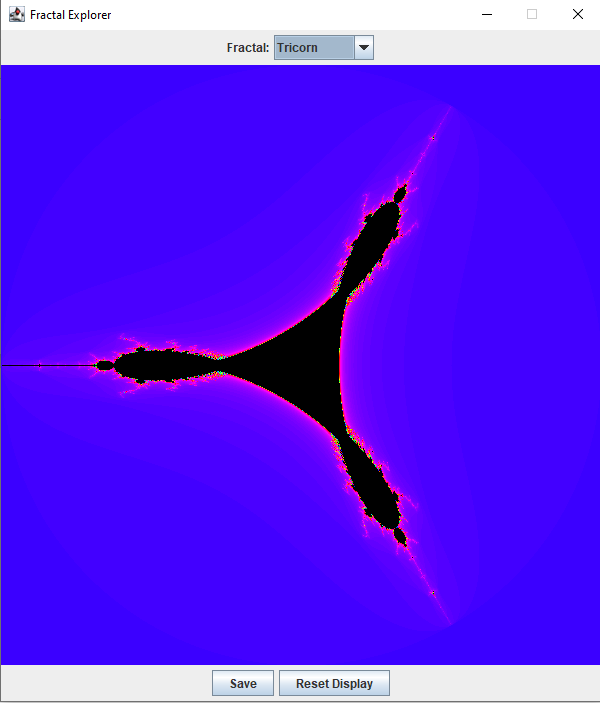
• В обработчике кнопки «Save Image» вам необходимо реализовать возможность указания пользователем, в какой файл он будет сохранять изображение.

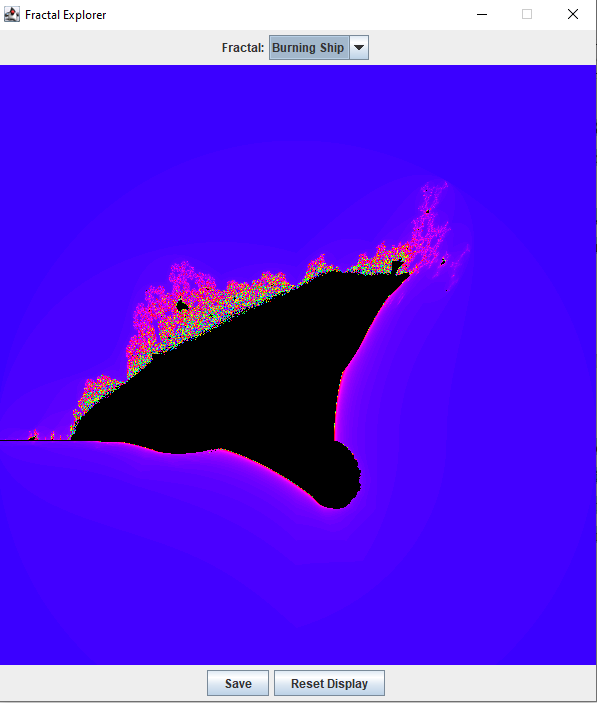
• Если пользователь успешно выбрал файл, следующим шагом является сохранения изображения фрактала на диск.

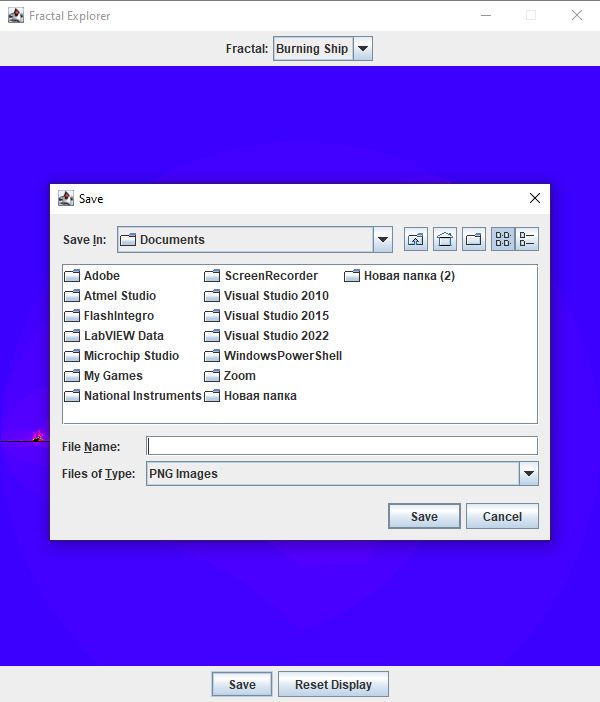
Ход работы

1. Результат работы кода.









Код добавленный в класс FractalExplorer

//Задание ComboBox  
JComboBox myComboBox = new JComboBox();  
  
//Добавление каждого объекта в поле со списком  
FractalGenerator mandelbrotFractal = new Mandelbrot();  
myComboBox.addItem(mandelbrotFractal);  
FractalGenerator tricornFractal = new Tricorn();  
myComboBox.addItem(tricornFractal);  
FractalGenerator burningShipFractal = new BurningShip();  
myComboBox.addItem(burningShipFractal);  
  
//Обработчик кнопок в поле со списком фракталов  
ButtonHandler fractalChooser = new ButtonHandler();  
myComboBox.addActionListener(fractalChooser);  
  
//Добавлеие подписи ComboBox и его расположение сверху  
JPanel myPanel = new JPanel();  
JLabel myLabel = new JLabel("Fractal:");  
myPanel.add(myLabel);  
myPanel.add(myComboBox);  
myframe.add(myPanel, BorderLayout.*NORTH*);  
  
// Создание кнопки для сохранения изображения фрактала и расположение снизу  
JButton saveButton = new JButton("Save");  
JPanel myBottomPanel = new JPanel();  
myBottomPanel.add(saveButton);  
myBottomPanel.add(resetButton);  
myframe.add(myBottomPanel, BorderLayout.*SOUTH*);  
  
//Обработчик кнопки сохранения  
ButtonHandler saveHandler = new ButtonHandler();  
saveButton.addActionListener(saveHandler);

private class ButtonHandler implements ActionListener  
{  
 public void actionPerformed(ActionEvent e)  
 {  
 //Выбор команды  
 String command = e.getActionCommand();  
  
 //Действия при работе с ComboBox, выбор фрактала и его отображение  
 if (e.getSource() instanceof JComboBox) {  
 JComboBox mySource = (JComboBox) e.getSource();  
 fractal = (FractalGenerator) mySource.getSelectedItem();  
 fractal.getInitialRange(range);  
 drawFractal();  
  
 }  
 //Действия при нажатии кнопки сброса  
 else if (command.equals("Reset")) {  
 fractal.getInitialRange(range);  
 drawFractal();  
 }  
 //Действия при нажатии кнопки сохранения  
 else if (command.equals("Save")) {  
  
 //Выбор сохранить изображение  
 JFileChooser myFileChooser = new JFileChooser();  
  
 //Сохранение в png формате  
 FileFilter extensionFilter =  
 new FileNameExtensionFilter("PNG Images", "png");  
 myFileChooser.setFileFilter(extensionFilter);  
 //Запрет на другие разрешения изображения  
 myFileChooser.setAcceptAllFileFilterUsed(false);  
  
 //Окно для выбора папки и сохранения изображения  
 int userSelection = myFileChooser.showSaveDialog(display);  
  
 //Действия при правильных действиях описанных выше  
 if (userSelection == JFileChooser.*APPROVE\_OPTION*) {  
  
 //Получение файла и его имени  
 java.io.File file = myFileChooser.getSelectedFile();  
 String file\_name = file.toString();  
  
 //Исключение при сохранении файла  
 try {  
 BufferedImage displayImage = display.getImage();  
 javax.imageio.ImageIO.*write*(displayImage, "png", file);  
 }  
  
 catch (Exception exception) {  
 JOptionPane.*showMessageDialog*(display,  
 exception.getMessage(), "Cannot Save Image",  
 JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
 //При отмене операции сохранения  
 else return;  
 }  
 }  
}

Код класса Tricorn

package com.company;  
import java.awt.geom.Rectangle2D;  
  
  
public class Tricorn extends FractalGenerator  
{  
  
 public static final int *MAX\_ITERATIONS* = 2000;  
  
  
 //определяет область комплексной плоскости для фрактала  
 public void getInitialRange(Rectangle2D.Double range)  
 {  
 range.x = -2;  
 range.y = -2;  
 range.width = 4;  
 range.height = 4;  
 }  
  
 //итеративная функция для фрактала  
 public int numIterations(double x, double y)  
 {  
 int iteration = 0;  
  
 double zreal = 0;  
 double zimaginary = 0;  
  
 while (iteration < *MAX\_ITERATIONS* &&  
 zreal \* zreal + zimaginary \* zimaginary < 4)  
 {  
 double zrealUpdated = zreal \* zreal - zimaginary \* zimaginary + x;  
 double zimaginaryUpdated = -2 \* zreal \* zimaginary + y;  
 zreal = zrealUpdated;  
 zimaginary = zimaginaryUpdated;  
 iteration += 1;  
 }  
  
  
 if (iteration == *MAX\_ITERATIONS*)  
 {  
 return -1;  
 }  
  
 return iteration;  
 }  
  
  
 public String toString() {  
 return "Tricorn";  
 }  
  
}

Код класса BurningShip

package com.company;  
  
import java.awt.geom.Rectangle2D;  
  
  
public class BurningShip extends FractalGenerator  
{  
  
 public static final int *MAX\_ITERATIONS* = 2000;  
  
  
 public void getInitialRange(Rectangle2D.Double range)  
 {  
 range.x = -2;  
 range.y = -2.5;  
 range.width = 4;  
 range.height = 4;  
 }  
  
  
 public int numIterations(double x, double y)  
 {  
  
 int iteration = 0;  
  
 double zreal = 0;  
 double zimaginary = 0;  
  
  
 while (iteration < *MAX\_ITERATIONS* &&  
 zreal \* zreal + zimaginary \* zimaginary < 4)  
 //абсолютное значение на каждой итерации  
 {  
 double zrealUpdated = zreal \* zreal - zimaginary \* zimaginary + x;  
 double zimaginaryUpdated = 2 \* Math.*abs*(zreal)  
 \* Math.*abs*(zimaginary) + y;  
  
 zreal = zrealUpdated;  
 zimaginary = zimaginaryUpdated;  
  
 iteration += 1;  
 }  
  
  
 if (iteration == *MAX\_ITERATIONS*)  
 {  
 return -1;  
 }  
  
 return iteration;  
 }  
  
  
 public String toString() {  
 return "Burning Ship";  
 }  
  
}

Вывод

В процессе выполнения данной лабораторной работы мы добавили в JAVA-приложение по поддержку нескольких фракталов и реализацию возможности выбора нужного фрактала из выпадающего списка и поддержку сохранения текущего изображения в файл

Ссылка на Github: https://github.com/ElenaGolubeva/Laboratory-5-Java

Список использованной литературы

1. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа,

2006.

1. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004.