

Московский Государственный Технический Университет
имени Н.Э. Баумана

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-3 «Информационные системы и телекоммуникации»

Отчет к лабораторной работе №4
по курсу «Технологии программирования»
“Углубленное ознакомление работы классов в языке программирования Java”

Продолжительность работы: 4 ак. часа

Сдал

Ситниченко С. А. ИУ3-42Б

Принял

Руденкова Ю.С.

Москва, 2022

Лабораторная №4

Тема: Программа для изучения фракталов

Цель работы: создание программы для вычисления и отрисовки фрактала с возможностью приближения и очистки экрана с использованием Swing Framework.

Создание пользовательского интерфейса

```
import java.awt.geom.Rectangle2D;
import java.awt.image.BufferedImage;

2 usages
public class JImageDisplay extends javax.swing.JComponent {
    8 usages
    private java.awt.image.BufferedImage image;

    1 usage
    public JImageDisplay(int wight, int height) {
        this.image = new BufferedImage(wight, height, BufferedImage.TYPE_INT_RGB);
        this.setPreferredSize(new Dimension(wight, height));
    }

    @Override
    protected void paintComponent(Graphics g) {
        g.drawImage(image, 0, 0, image.getWidth(), image.getHeight(), observer: null);
    }

    public void clearImage() {
        for (int i = 0; i < image.getHeight(); i++) {
            for (int j = 0; j < image.getWidth(); j++) {
                image.setRGB(i, j, 0);
            }
        }
    }

    2 usages
    public void drawPixel(int x, int y, int rgbColor) {
        image.setRGB(x, y, rgbColor);
    }
}
```

Рис. 1 Класс JImageDisplay.java

Класс `JImageDisplay` управляет изображением. Конструктор `JImageDisplay` получает целые ширину и высоту и инициализирует поле `BufferedImage` ссылкой на новое изображение заданной ширины и высоты.

Также он вызывает метод родительского класса `setPreferredSize()` передавая ему заданные ширину и высоту. Тогда создаваемый компонент начинает прорисовывать изображение после того как будет добавлен к пользовательскому интерфейсу.

Метод `clearImage()` устанавливает черный цвет для всех пикселей изображения, а `drawPixel(int x, int y, int rgbColor)` - задает указанный цвет пиксела.

Вычисление фрактала Мандельброта

```
public class Mandelbrot extends FractalGenerator {  
    2 usages  
    public static final int MAX_ITERATIONS = 2000;  
  
    2 usages  
    @Override  
    public void getInitialRange(Rectangle2D.Double range) {  
        double newWidth = range.width;  
        double newHeight = range.height;  
  
        range.x = -2;  
        range.y = -1.5;  
        range.width = 3;  
        range.height = 3;  
    }  
  
    1 usage  
    @Override  
    public int numIterations(double x, double y) {  
        int iterator = 0;  
        double real = 0;  
        double imaginary = 0;  
  
        while (iterator < MAX_ITERATIONS && real * real + imaginary * imaginary < 4) {  
            double newReal = real * real - imaginary * imaginary + x;  
            double newImaginary = 2 * real * imaginary + y;  
            real = newReal;  
            imaginary = newImaginary;  
            iterator++;  
        }  
  
        if (iterator == MAX_ITERATIONS) {  
            return -1;  
        }  
        return iterator;  
    }  
}
```

Рис. 2 класс Mandelbrot

В классе Mandelbrot происходит вычисление фрактала Мандельброта по функции $z_n = z_{n-1}^2 + c$.

Метод `getInitialRange(Rectangle2D.Double)` позволяет указать генератору фрактала, какая часть комплектной области "интересна" для вычисления фрактала.

Метод `numIterations(double, double)` содержит реализацию итеративной функции фрактала Мандельброта.

Константа `MAX_ITERATIONS` определена для указания максимального числа итераций.

GUI и отрисовка фрактала, класс `FractalExplorer`

```
public void createAndShowGUI() {  
    JFrame frame = new JFrame( title: "Fractal");  
    frame.add(display, BorderLayout.CENTER);  
  
    JButton button = new JButton( text: "Reset Display");  
  
    ButtonHandler resetHandler = new ButtonHandler();  
    button.addActionListener(resetHandler);  
  
    MouseHandler click = new MouseHandler();  
    display.addMouseListener(click);  
  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    JPanel bottomPanel = new JPanel();  
    bottomPanel.add(button);  
    frame.add(bottomPanel, BorderLayout.SOUTH);  
  
    frame.pack();  
    frame.setVisible(true);  
    frame.setResizable(false);  
}
```

Рис. 3 метод `CreateAndShowGUI`

Данный метод инициализирует графический пользовательский интерфейс Swing: объект `JFrame` содержащий объект `JImageDisplay` и кнопку для сброса экрана в начальное состояние.

```

private void drawFractal() {
    for (int x = 0; x < windowSize; x++) {
        for (int y = 0; y < windowSize; y++) {

            double xCoord = fractal.getCoord(rect.x, rangeMax: rect.x + rect.width, windowSize, x);
            double yCoord = fractal.getCoord(rect.y, rangeMax: rect.y + rect.height, windowSize, y);

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

            if (iteration == -1){
                display.drawPixel(x, y, rgbColor: 0);
            }

            else {
                float hue = 0.6f + (float) iteration / 200f;
                int rgbColor = Color.HSBtoRGB(hue, saturation: 1f, brightness: 1f);
                display.drawPixel(x, y, rgbColor);
            }
        }
    }

    display.repaint();
}

```

Рис. 4 метод drawFractal

Метод отвечает за отрисовку фрактала относительно координат окна и задание его цвета.

```

private class ButtonHandler implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();

        if (command.equals("Reset Display")) {
            fractal.getInitialRange(rect);
            drawFractal();
        }
    }
}

2 usages
private class MouseHandler extends MouseAdapter {
    public void mouseClicked(MouseEvent e) {
        int x = e.getX();
        double xCoord = fractal.getCoord(rect.x,
            rangeMax: rect.x + rect.width, windowSize, x);

        int y = e.getY();
        double yCoord = fractal.getCoord(rect.y,
            rangeMax: rect.y + rect.height, windowSize, y);

        fractal.recenterAndZoomRange(rect, xCoord, yCoord, scale: 0.5);

        drawFractal();
    }
}

```

Рис. 5 хэндлеры

Данные методы отвечают за обработку такого события как нажатие мышью на кнопку.

```

public static void main(String[] args)
{
    FractalExplorer displayExplorer = new FractalExplorer( size: 600);
    displayExplorer.createAndShowGUI();
    displayExplorer.drawFractal();
}

```

Рис. 6 main

Результат

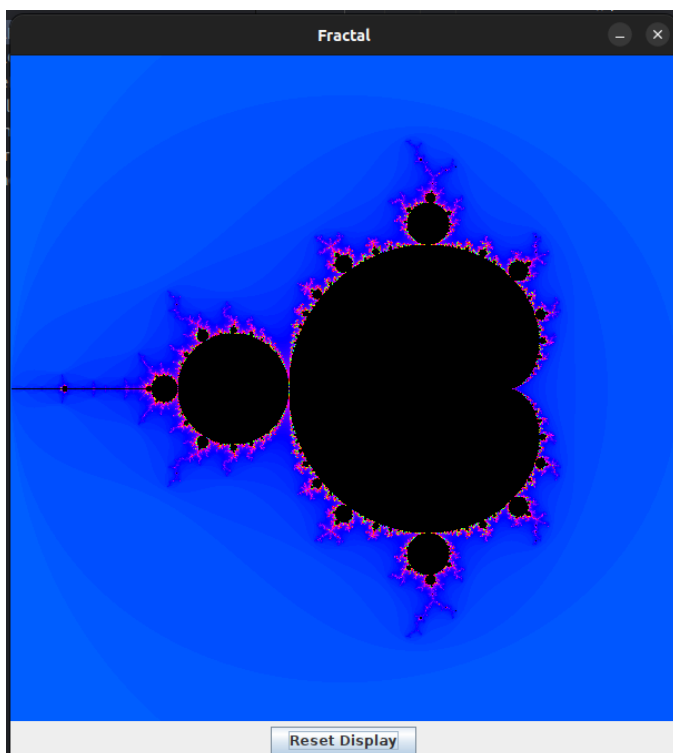


Рис. 7 результат

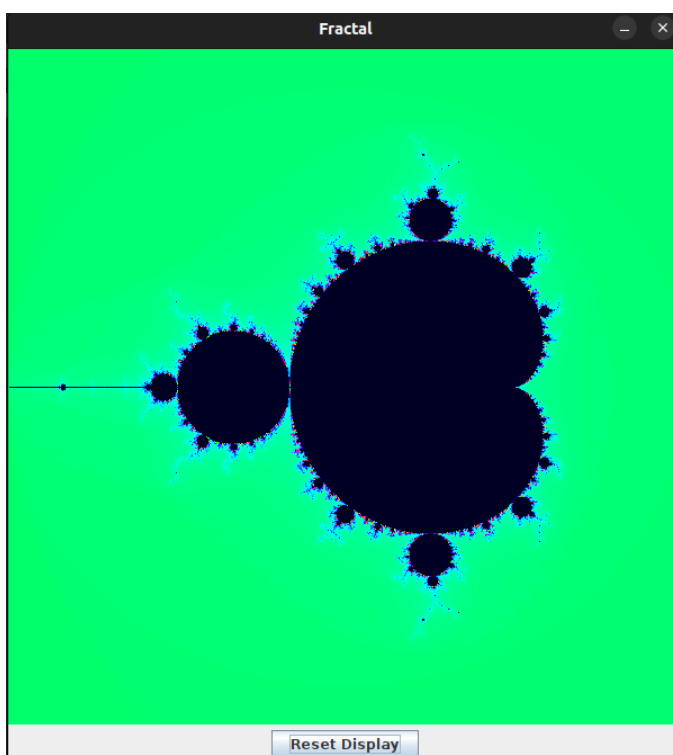


Рис. 8 доп. задание

Вывод:

Была разработана программа для вычисления и отрисовки фрактала с возможностью приближения и очистки экрана с использованием Swing Framework. В процессе работы были применены принципы ООП.