**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Информатики

****

**Отчет по лабораторной работе № 4**

по дисциплине «КТП»

на тему:

«**Рисование фракталов»**

Выполнила: студентка группы БВТ1802

Лаврухина Елена Павловна

Руководитель:

Полянцева Ксения Андреевна

Москва 2020

Цель работы

Создать небольшое JAVA-приложение, которое сможет рисовать фракталы. Это должно быть сделано с помощью фреймворка Swing и Java API, который позволяет создавать графические пользовательские интерфейсы.

Выполнение

Код программы

1. JImageDisplay

**import** javax.swing.\*;  
**import** java.awt.image.\*;  
**import** java.awt.\*;  
 *//наследование от класса JComponent***public class** JImageDisplay **extends** JComponent {  
 *//для управления изображением* **private** BufferedImage **image**;  
 *//новые значения длины и ширины изображения и его тип* **public** JImageDisplay(**int** width, **int** height) {  
 **image** = **new** BufferedImage(width, height, BufferedImage.***TYPE\_INT\_RGB***);  
 Dimension imageDimension = **new** Dimension(width, height);  
 *//метод, который включит наш компонент в пользовательский интерфейс* **super**.setPreferredSize(imageDimension);  
 }  
 *//отрисовка* **public void** paintComponent (Graphics g) {  
 **super**.paintComponent(g);  
 g.drawImage (**image**, 0, 0, **image**.getWidth(), **image**.getHeight(), **null**);  
 }  
 *//установка всех пикселей изображения в черный цвет* **public void** clearImage() {  
 **for** (**int** i=0;i<**image**.getWidth();i++)  
 **for** (**int** j=0;j<**image**.getHeight();j++)  
 **image**.setRGB(i,j,0);  
 }  
 *//установка пикселя в определенный цвет* **public void** drawPixel(**int** x, **int** y, **int** rgbColor) {  
 **image**.setRGB(x,y,rgbColor);  
 }  
}

1. Mandelbrot

**import** java.awt.geom.Rectangle2D;  
 *// подкласс FractalGenerator***public class** Mandelbrot **extends** FractalGenerator {  
 *//константа с максимальным количеством итераций* **public static final int *MAX\_ITERATIONS*** = 2000;  
 *//начальный диапазон* **public void** getInitialRange(Rectangle2D.Double range) {  
 range.**x** = -2;  
 range.**y** = -1.5;  
 range.**width** = 3;  
 range.**height** = 3;  
 }  
 *//Вычисляем количество итераций с помощью мнимой и реальной части* **public int** numIterations(**double** x, **double** y) {  
 **int** iterations = 0;  
 **double** zreal = 0;  
 **double** zimaginary = 0;  
  
**while** (iterations < ***MAX\_ITERATIONS*** && zreal \* zreal + zimaginary \* zimaginary < 4) {  
 **double** zreal1 = zreal \* zreal - zimaginary \* zimaginary + x;  
 **double** zimaginary1 = 2 \* zreal \* zimaginary + y;  
 zreal = zreal1;  
 zimaginary = zimaginary1;  
 iterations += 1;  
 } *//Если достигнут максимум, вернуть -1 (точка не вышла за пределы границы)* **if** (iterations == ***MAX\_ITERATIONS***) {  
 **return** -1;  
 }  
 **return** iterations;  
 }  
}

1. FractalExplorer

**import** java.awt.\*;  
**import** javax.swing.\*;  
**import** java.awt.geom.Rectangle2D;  
**import** java.awt.event.\*;  
  
**public class** FractalExplorer {  
 **private int displaySize**;  
 **private** JImageDisplay **display**;  
 **private** FractalGenerator **fractal**;  
 **private** Rectangle2D.Double **range**;  
  
 **public** FractalExplorer(**int** size) {  
 **displaySize** = size;  
 **fractal** = **new** Mandelbrot();  
 **range** = **new** Rectangle2D.Double();  
 **fractal**.getInitialRange(**range**);  
 **display** = **new** JImageDisplay(**displaySize**, **displaySize**);  
 }  
 **public void** createAndShowGUI() {  
 **display**.setLayout(**new** BorderLayout());  
 JFrame frame = **new** JFrame(**"Fractal Explorer"**);  
 *//позиция center* frame.add(**display**, BorderLayout.***CENTER***);  
 *//Кнопка сброса* JButton resetButton = **new** JButton(**"Reset"**);  
 *//обработчик сброса* ResetHandler handler = **new** ResetHandler();  
 resetButton.addActionListener(handler);  
 *// позиция south* frame.add(resetButton, BorderLayout.***SOUTH***);  
 *//обработчик для мыши* MouseHandler click = **new** MouseHandler();  
 **display**.addMouseListener(click);  
 *//закрытие окна* frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  
 frame.pack();  
 *//видимость изображения* frame.setVisible(**true**);  
 *//запрет изменения размера* frame.setResizable(**false**);  
 }  
 **private void** drawFractal() {  
 *//проходим через каждый пиксель на дисплее* **for** (**int** x=0; x<**displaySize**; x++) {  
 **for** (**int** y=0; y<**displaySize**; y++) {  
 *//находим координаты* **double** xCoord = **fractal**.*getCoord*(**range**.**x**, **range**.**x** + **range**.**width**, **displaySize**, x);  
 **double** yCoord = **fractal**.*getCoord*(**range**.**y**, **range**.**y** + **range**.**height**, **displaySize**, y);  
 *//количество итерации* **int** iterations = **fractal**.numIterations(xCoord, yCoord);  
 *//черный цвет* **if** (iterations == -1) {  
 **display**.drawPixel(x, y, 0);  
 }  
 **else** { *//выбрать цвет в зависимости от итерации* **float** hue = 0.7f + (**float**) iterations / 200f;  
 **int** rgbColor = Color.*HSBtoRGB*(hue, 1f, 1f);  
 *//обновляем цвет* **display**.drawPixel(x, y, rgbColor);  
 }  
 }  
 } *//обновляем изображение* **display**.repaint();  
 }  
 **private class** ResetHandler **implements** ActionListener {  
 *//сбрасываем в начало и рисуем фрактал* **public void** actionPerformed(ActionEvent e) {  
 **fractal**.getInitialRange(**range**);  
 drawFractal();  
 }  
 }  
 **private class** MouseHandler **extends** MouseAdapter {  
 **public void** mouseClicked(MouseEvent e) {  
 *//Получение координат в области щелчка мыши* **int** x = e.getX();  
 **double** xCoord = **fractal**.*getCoord*(**range**.**x**, **range**.**x** + **range**.**width**, **displaySize**, x);  
 **int** y = e.getY();  
 **double** yCoord = **fractal**.*getCoord*(**range**.**y**, **range**.**y** + **range**.**height**, **displaySize**, y);  
 **fractal**.recenterAndZoomRange(**range**, xCoord, yCoord, 0.5);  
 *//перерисовываем фрактал* drawFractal();  
 }  
 }  
 **public static void** main(String[] args) {  
 FractalExplorer displayExplorer = **new** FractalExplorer(800);  
 displayExplorer.createAndShowGUI();  
 displayExplorer.drawFractal();  
 }  
}

Скриншоты работы программы

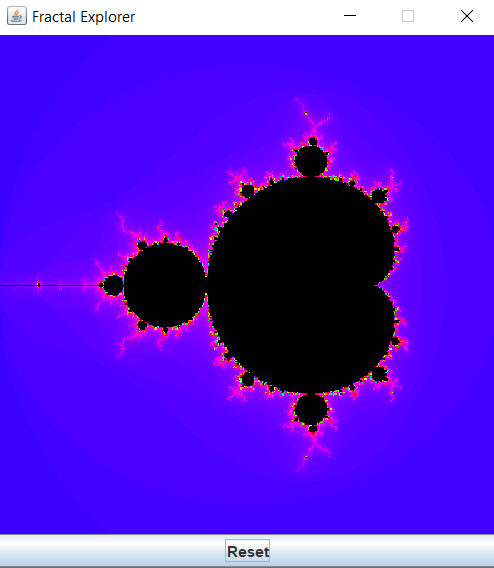


Рисунок 1 – Работа программы

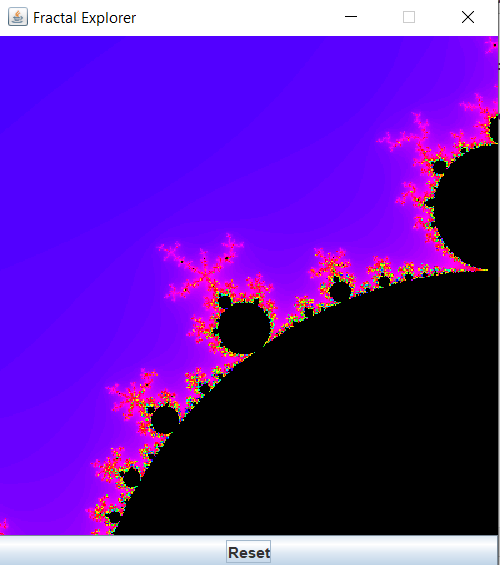


Рисунок 2 – Работа программы

Вывод

В ходе работы мы создали небольшое JAVA-приложение, которое может рисовать фракталы. Оно сделано с помощью фреймворка Swing и Java API.