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 «Информационные технологии»

Лабораторная работа №4:   
«Рисование фракталов**»**

Выполнил: студент гр. БПЗ1902

Оглоблин Д. И.

Проверил: Харрасов К.Р.

Москва 2021 г.

Цель работы:

В данной лабораторной работе необходимо создать небольшое JAVA-приложение, которое сможет рисовать фракталы.

Задание:

• Класс FractalExplorer должен отслеживать несколько важных полей для состояния программы:

1) Целое число «размер экрана», которое является шириной и высотой отображения в пикселях. (Отображение фрактала будет квадратным.)   
 2) Ссылка JImageDisplay, для обновления отображения в разных методах в процессе вычисления фрактала.   
 3) Объект FractalGenerator. Будет использоваться ссылка на базовый класс для отображения других видов фракталов в будущем.   
 4) Объект Rectangle2D.Double, указывающий диапазона комплексной плоскости, которая выводится на экран.

• У класса должен быть конструктор, который принимает значение размера отображения в качестве аргумента, затем сохраняет это значение в соответствующем поле, а также инициализирует объекты диапазона и фрактального генератора. Данный конструктор не должен устанавливать какие-либо компоненты Swing; они будут установлены в следующем методе.

• Создайте метод createAndShowGUI (), который инициализирует графический интерфейс Swing: JFrame, содержащий объект JimageDisplay, и кнопку для сброса отображения. Используйте java.awt.BorderLayout для содержимого окна; добавьте объект отображения изображения в позицию BorderLayout.CENTER и кнопку в позицию BorderLayout.SOUTH.

• Реализуйте вспомогательный метод с типом доступа private для вывода на экран фрактала, можете дать ему имя drawFractal ().

• Создайте внутренний класс для обработки событий java.awt.event.ActionListener от кнопки сброса. Обработчик должен сбросить диапазон к начальному, определенному генератором, а затем перерисовать фрактал.

• Создайте другой внутренний класс для обработки событий java.awt.event.MouseListener с дисплея. Вам необходимо обработать события от мыши, поэтому вы должны унаследовать этот внутренний класс от класса MouseAdapterAWT.

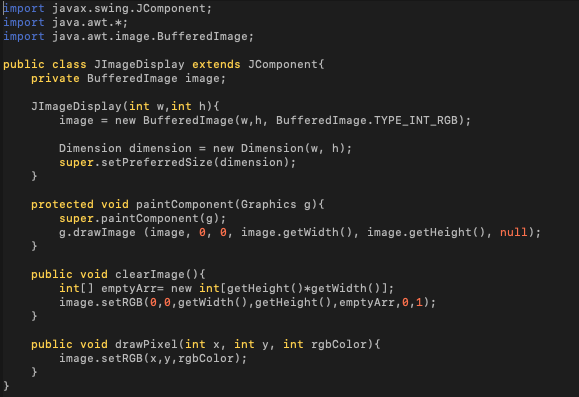
• В заключении, вам необходимо создать статический метод main() для FractalExplorer так, чтобы можно было его запустить.

Выполнение работы:

1. Создание пользовательского интерфейса:

Прежде чем рисовать фракталы, необходимо создать графический виджет, который будет их отображать.

• Создайте класс JImageDisplay. Класс BufferedImage управляет изображением, содержимое которого можно записать.

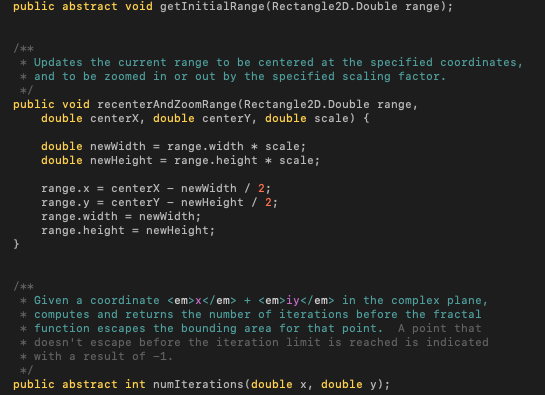


2. Код для вычисления фрактала Мандельброта.

Для создания фракталов используем следующий исходный файл FractalGenerator.java, от которого будут унаследованы все фрактальные генераторы.

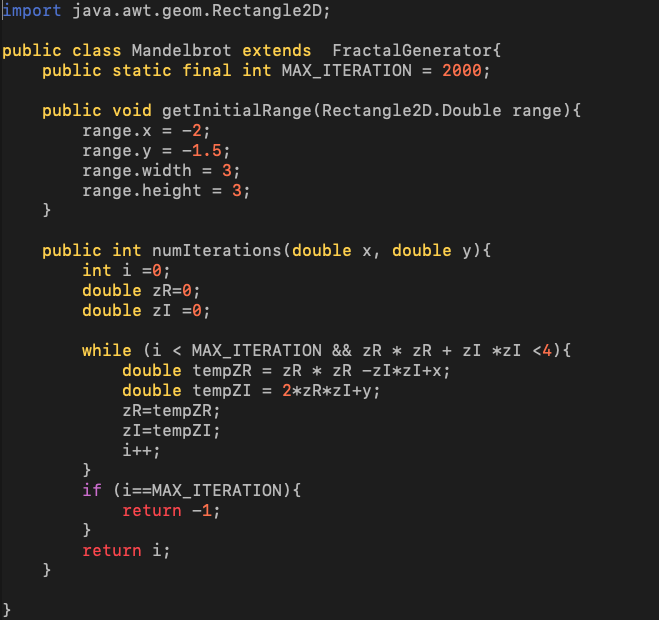


getInitialRange (Rectangle2D.Double) - метод позволяет генератору фракталов определить наиболее «интересную» область комплексной плоскости для конкретного фрактала.

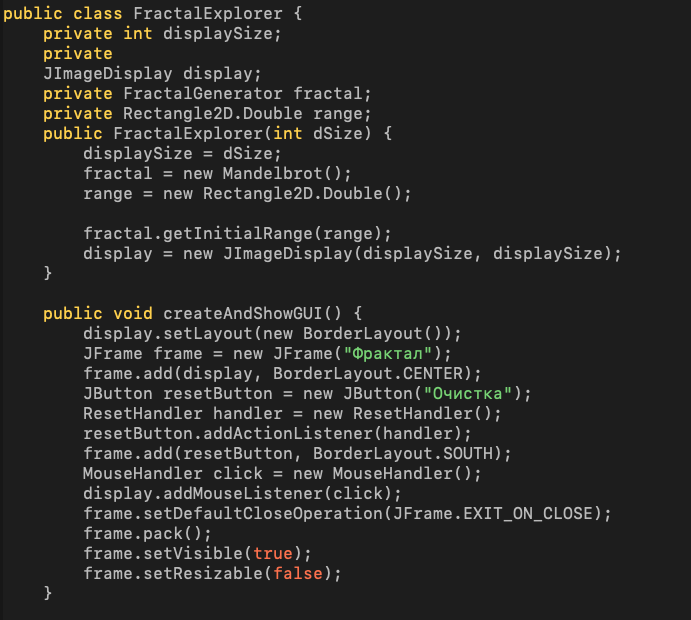


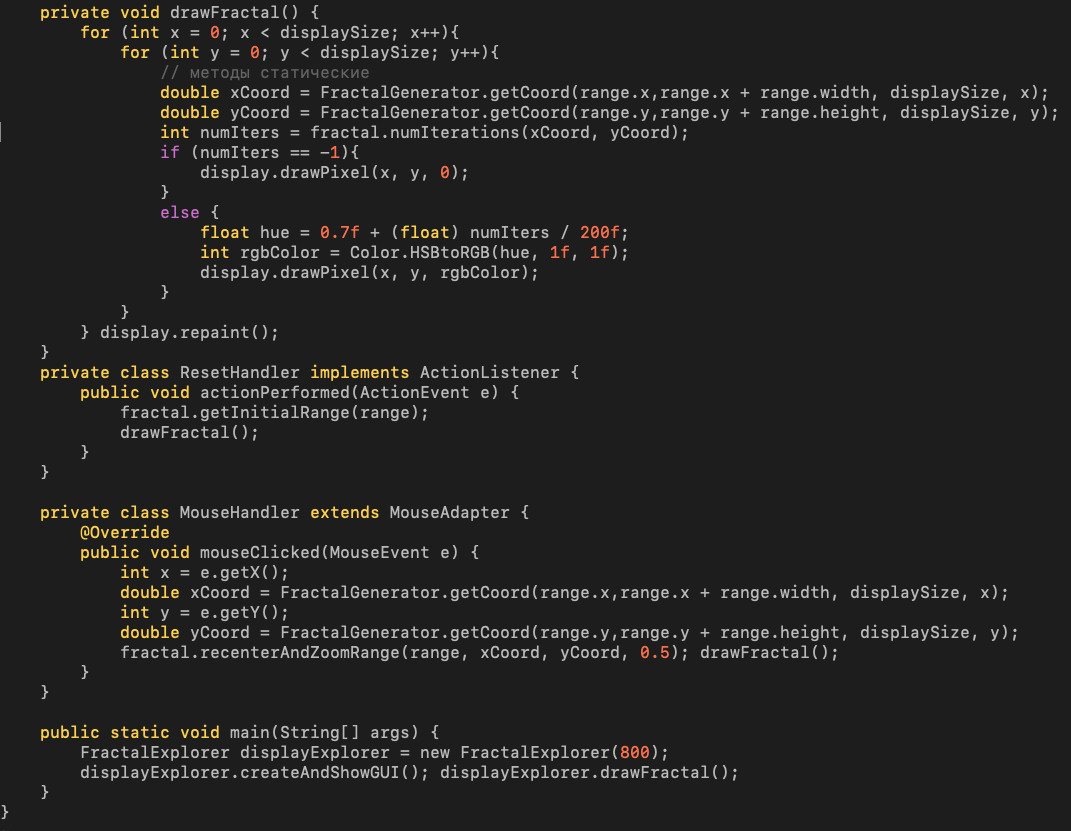
Метод numIterations(double, double) реализует итеративную функцию для фрактала Мандельброта

Создайте подкласс FractalGenerator с именем Mandelbrot. в нем вам необходимо будет обеспечить только два метода: getInitialRange() и numIterations().



3. Создание класса FractalExplorer, который позволит исследовать различные области фрактала, путем его создания, отображения через графический интерфейс Swing и обработки событий, вызванных взаимодействием приложения с пользователем.





Результат работы:

Если число итераций равно -1 (т.е. точка не выходит за границы, установите пиксель в черный цвет (для rgb значение 0). Иначе выберите значение цвета, основанное на количестве итераций. Можно также для этого использовать цветовое пространство HSV: поскольку значение цвета варьируется от 0 до 1, получается плавная последовательность цветов от красного к желтому, зеленому, синему, фиолетовому и затем обратно к красному! Для этого вы можете использовать следующий фрагмент: float hue = 0.7f + (float) numIters / 200f; int rgbColor = Color.HSBtoRGB(hue, 1f, 1f);

