**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 № 6

по дисциплине «Информационные технологии и программирование»

Выполнил: студент группы БФИ2102

Голубева Елена Александровна

Проверила:

Мосева Марина Сергеевна

Москва, 2022

Цель работы

Реализация в JAVA-приложении возможность рисования фрактала с несколькими фоновыми потоками.

Задание

Класс javax.swing.SwingWorker, который облегчает процесс организации фонового потока. SwingWorker - абстрактный класс, включающий в себя следующие важные методы:

• doInBackground() - метод, который фактически выполняет фоновые операции. Swing вызывает этот метод в фоновом потоке, а не в потоке обработке событий.

• done() - этот метод вызывается, когда фоновая задача завершена.

1) Создайте подкласс SwingWorker с именем FractalWorker, который будет внутренним классом FractalExplorer.

2) Класс FractalWorker будет отвечать за вычисление значений цвета для одной строки фрактала, поэтому ему потребуются два поля: целочисленная yкоордината вычисляемой строки, и массив чисел типа int для хранения вычисленных значений RGB для каждого пикселя в этой строке. Конструктор должен будет получать y-координату в качестве параметра и сохранять это.

3) Метод doInBackground() вызывается в фоновом потоке и отвечает за выполнение длительной задачи.

4) Вместо этого выделите память для массива целых чисел в начале реализации этого метода, а затем сохраните цвет каждого пикселя в этом массиве.

Метод doInBackground() должен возвращать объект типа Object, так как это указано в объявлении SwingWorker . Просто верните null.

5) Метод done() вызывается, когда фоновая задача завершена, и этот метод вызывается из потока обработки событий Swing.

Использовать метод JComponent.repaint(), который позволит вам указать область для перерисовки.

Для каждой строки в отображении создать отдельный рабочий объект, а затем вызвать для него метод execute ().

Добавить поле «rows remaining» в класс Fractal Explorer и использовать его, чтобы узнать, когда будет завершена перерисовка.

• Создайте функцию void enableUI(boolean val), которая будет включать или отключать кнопки с выпадающим списком в пользовательском интерфейсе на основе указанного параметра.

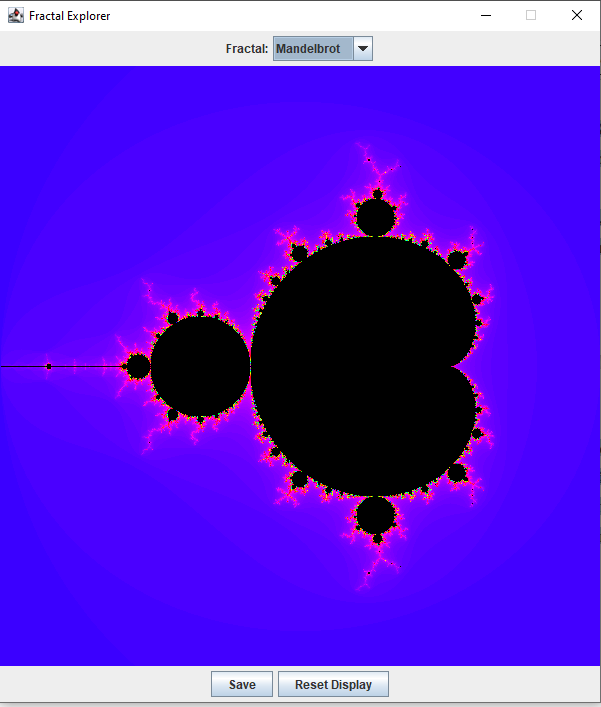
• Функция «draw fractal» должна сделать еще две вещи. Первая - она должна вызвать метод enableUI (false), чтобы отключить все элементы пользовательского интерфейса во время рисования. Вторая - она должна установить значение «rows remaining» равным общему количеству строк, которые нужно нарисовать.

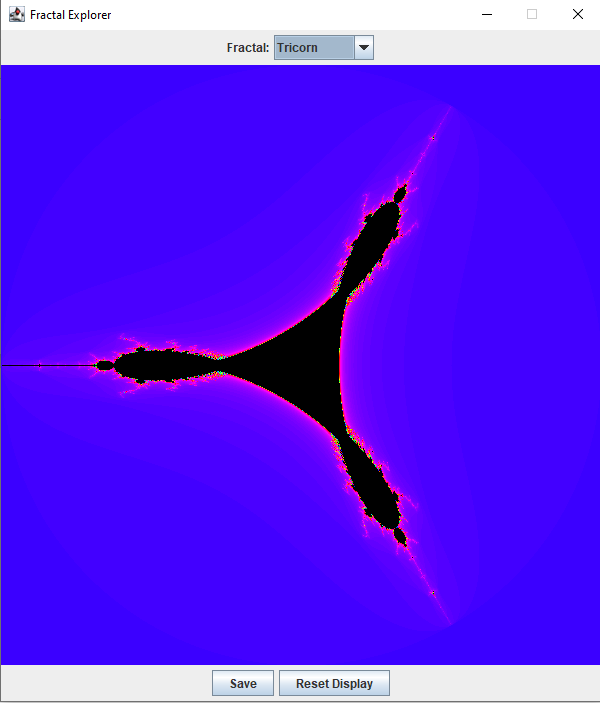
• В методе done(), уменьшите значение «rows remaining» на 1, как последний шаг данной операции. Затем, если после уменьшения значение «rows remaining» равно 0, вызовите метод enableUI (true).

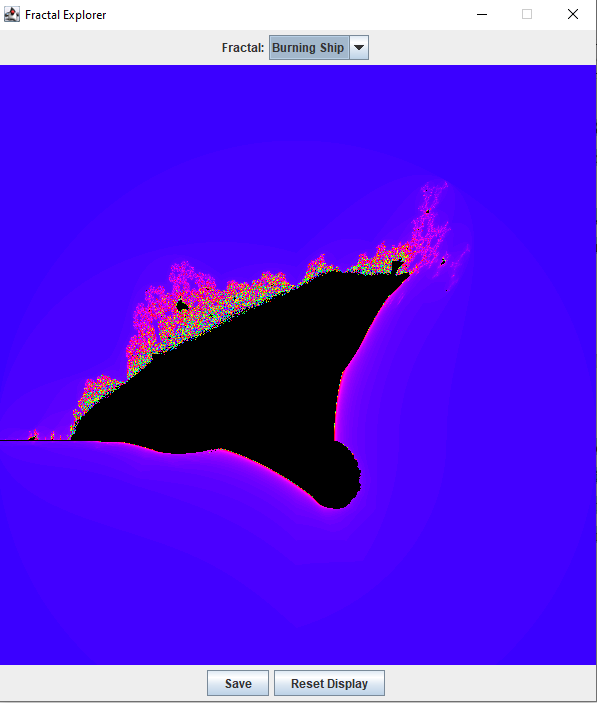
• Наконец, измените реализацию mouse-listener для того, чтобы она сразу возвращалась в предыдущее состояние, если значение «rows remaining» не равно нулю.

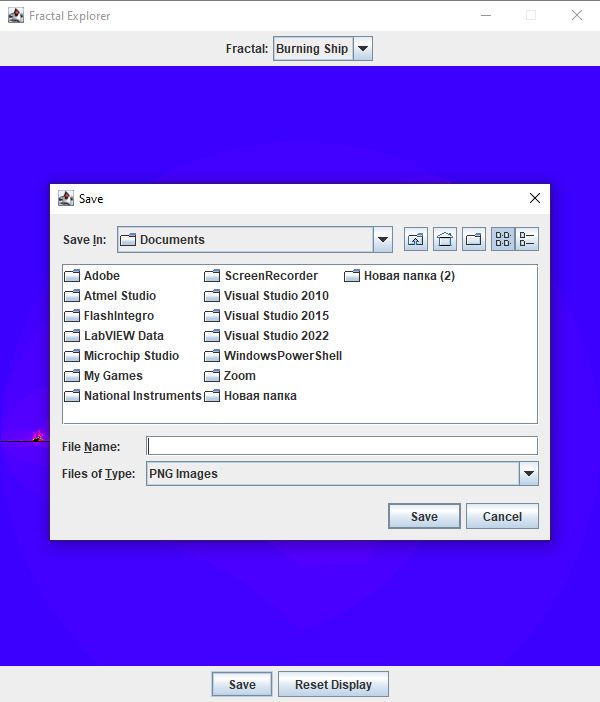
Ход работы

1. Результат работы кода.









Код добавленный в класс FractalExplorer

//метод для вывода фрактала на экран  
 private void drawFractal()  
 {  
 // При отрисовке фрактала кнопки не работают  
 enableUI(false);  
  
 rowsRemaining = displaySize;  
  
 //Перебор каждой строки на дисплее  
 for (int x=0; x<displaySize; x++){  
 //Отрисовка фрактала  
 FractalWorker drawRow = new FractalWorker(x);  
 drawRow.execute();  
 }  
  
 }  
  
//  
 //Отвечает за работу кнопок в зависиости от переданного значения  
 private void enableUI(boolean val) {  
 myComboBox.setEnabled(val);  
 resetButton.setEnabled(val);  
 saveButton.setEnabled(val);  
 }

//Вычисляет значения цвета для одной строки фрактала  
private class FractalWorker extends SwingWorker<Object, Object>{  
 //поля для координаты строки и массив для вычисленных значений  
 int yCoordinate;  
 int[] computedRGBValues;  
  
 //конструктор для сохранеия у  
 private FractalWorker(int row) {  
 yCoordinate = row;  
 }  
  
 //Вызывается в фоновом потоке, вычисляет значения цвета и сохраняет в массив  
 protected Object doInBackground() {  
  
 computedRGBValues = new int[displaySize];  
  
 // Цикл для всех пикселей в одной строке  
 for (int i = 0; i < computedRGBValues.length; i++) {  
  
 double xCoord = fractal.*getCoord*(range.x,  
 range.x + range.width, displaySize, i);  
 double yCoord = fractal.*getCoord*(range.y,  
 range.y + range.height, displaySize, yCoordinate);  
  
 // вычисление количества итераций для данной координаты  
 int iteration = fractal.numIterations(xCoord, yCoord);  
  
 // Установка черного цвета  
 if (iteration == -1){  
 computedRGBValues[i] = 0;  
 }  
  
 else {  
 // Цвет ыбирается в зависимости от количества итераций  
 float hue = 0.7f + (float) iteration / 200f;  
 int rgbColor = Color.*HSBtoRGB*(hue, 1f, 1f);  
  
 // Обновление массива для текущего пикселя  
 computedRGBValues[i] = rgbColor;  
 }  
 }  
 return null;  
  
}  
//Вызывается при завершении фоновой задачи. Отрисовывает строку и обновляет отображение  
protected void done() {  
 //  
 for (int i = 0; i < computedRGBValues.length; i++) {  
 display.drawPixel(i, yCoordinate, computedRGBValues[i]);  
 }  
 display.repaint(0, 0, yCoordinate, displaySize, 1);  
  
 // Уменьшение количества оставшихся строк  
 rowsRemaining--;  
 if (rowsRemaining == 0) {  
 enableUI(true);  
 }  
}  
  
}

Вывод

В процессе выполнения данной лабораторной работы мы реализовали в JAVA-приложении возможность рисования фрактала с несколькими фоновыми потоками и игнорирование нажатия кнопок до завершения отрисовки фрактала.

Ссылка на Github: https://github.com/ElenaGolubeva/Laboratory-6-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.