ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский Технический Университет Связи и Информатики»  
(МТУСИ)

Кафедра математической кибернетики и информационных технологий

Задача № 5: Выбор и сохранение фракталов

Выполнила:

Студентка 2 курса

Группы БСТ-1602

Феоктистова Лидия

Москва, 2018

**Задача:**

Расширить генератор фракталов двумя новыми функциями, научиться работать со множеством фракталов, выбирая их с помощью выпадающего списка и сохранять текущее изображение фрактала в файл.

**Анализ предметной области, выбор инструментария:**

Я использовала Java Development Kit (JDK) - комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), а для написания программы - программу Notepad++.

**Объяснение функций:**

1. class JImageDisplay

Метод **public void clearImage()** устанавливает все пиксели в изображении к черному цвету;

Метод **public void drawPixel(int x, int y, int rgbColor)** устанавливает определенный цвет для пикселя;

Метод **public BufferedImage getim()** возвращает рендерингованное изображение;

Метод **public void paintComponent( Graphics g)** выводит на экран данное изображение;

1. class Mandelbrot

Метод **public void getInitialRange(Rectangle2D.Double rec)** указывает генератору фрактала, какая часть комплексной области "интересна" для вычисления фрактала;

Метод **public int numIterations(double x, double y)** является реализацией итеративной функции фрактала Мандельброта;

Метод **public String toString()** возвращает название фрактала;

1. class Tricorn

Метод **public void getInitialRange(Rectangle2D.Double rec)** указывает генератору фрактала, какая часть комплексной области "интересна" для вычисления фрактала;

Метод **public int numIterations(double x, double y)** является реализацией итеративной функции фрактала Трикорн;

Метод **public String toString()** возвращает название фрактала;

1. class BurningShip

Метод **public void getInitialRange(Rectangle2D.Double rec)** указывает генератору фрактала, какая часть комплексной области "интересна" для вычисления фрактала;

Метод **public int numIterations(double x, double y)** является реализацией итеративной функции фрактала “Горящий корабль”;

Метод **public String toString()** возвращает название фрактала;

1. class FractalGenerator

Метод **public static double getCoord(double rangeMin, double rangeMax, int size, int coord)** преобразует целочисленные координаты в значение double;

Метод **public void recenterAndZoomRange(Rectangle2D.Double range, double centerX, double centerY, double scale)** обновляет текущий диапазон для указанного центра;

Метод **public abstract void getInitialRange(Rectangle2D.Double range)** задает указанный прямоугольник;

Метод **public abstract int numIterations(double x, double y)** Вычисляет количество итераций для фрактала, если достигает предела, возвращает -1;

1. class FractalExplorer

Метод **void createAndShowGUI()** создает и реализует окно с фракталом;

Метод **void drawFractal()** рисует фрактал;

Класс **mouse\_listener** обрабатывает события MouseListener (обработка щелчка мыши);

Метод **public void mouseClicked(MouseEvent e)** преобразует координаты точки изображения в отображаемую область фрактала;

Класс **act\_listener** обрабатывает события ActionListener от кнопки сброса и сохранения;

Метод **public void actionPerformed(ActionEvent e)** сбрасывает диапазон отображения в начальное значение или сохраняет текущее изображение фрактала;

Главный метод **public static void main(String[] args)** запускает окно и рисует фрактал.

**Выводы:**

Я написала небольшое JAVA-приложение, которое рисует фрактал, выбранный из выпадающего списка. Приближение осуществляется щелчком мыши по изображению фрактала, сброс – нажатием кнопки Reset Display, а сохранение текущего изображения фрактала – кнопкой Save Image.

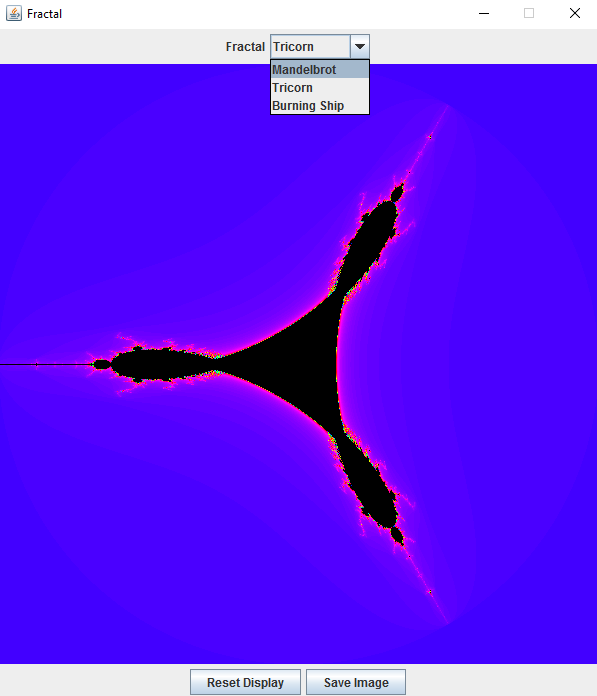


Рис. Полученное приложение. Выбран фрактал Tricorn.