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

Лабораторная работа №6

# **«Многоядерный исследователь/генератор фракталов»**

Выполнил:

Студент 2 курса

группы БСТ1602

Чернюк Александр

Проверил:

Городничев М. Г.

Москва 2018

Оглавление

[**«Фрактал Эксплорер»** 1](#_Toc512863935)

[Цель лабораторной работы: 2](#_Toc512863936)

[Задачи: 2](#_Toc512863937)

[Анализ предметной области и выбор инструментария: 2](#_Toc512863938)

[Ход решения задачи: 2](#_Toc512863939)

[Вывод: 3](#_Toc512863940)

## Цель лабораторной работы:

Получить общее представление о создании программ на языке Java, ознакомиться с синтаксисом и структурой программ, узнать, как пишется приложение, отрисовывающее фракталы. Доработать класс FractalExplorer, реализовав проверку точек на идентичность и реализовав метод кучи. Добавить 2 фрактала: Burning Ship, Tricorn. Создать выпадающий список, где можно будет выбрать нужный фрактал, реализовать рисование в фоновом режиме с помощью SwingWorker, создать возможность сохранения фракталов в виде .png.

# Задачи:

Написать программу, которая рисует фрактал, увеличивающий свой масштаб при нажатии на него.

# Анализ предметной области и выбор инструментария:

Я использовал среду разработки Eclipse, потому что IntelliJ IDEA запросила ключ активации.

# Ход решения задачи:

1. Создаем класс JImageDisplay. Он содержит несколько функций и конструктор, таких как:

Конструктор JImageDisplay(), который получает ширину и высоту и инициализирует ими переменную mImage.

PaintComponent() – для того, чтобы отрисовать изображение.

clearImage() – устанавливает все пиксели изображения на черный цвет.

drawPixel – установка пикселю на конкретной координате цвета

1. Создаем класс Mandelbrot. Он содержит несколько функций, таких как:

getInitialRange() позволяет определять какая часть плоскости является самой перспективной для фрактала.

numIterations() Высчитывает число итераций для данного фрактала: на протяжении 2000 итераций или пока z1 < 4, считает переменную count, которая и является числом нужных итераций.

1. Создаем класс FractalExplorer.

createAndShowGUI() создает интерфейс юзера: делает иллюстрацию кнопку сброса, которая обрабатывается командой «reset». Делает фрэйм с заглавием.

FractalExplorer() конструктор, который инициализирует размер дисплея и иные переменные, в частности сообщает, какой класс фрактала рисовать на иллюстрации.

mouseListener() разрешает включить перечень возможностей мышки, в предоставленной истории оверрайдит нажатие кнопки мыши, дабы «увеличивать» иллюстрацию фрактала. Мы элементарно увеличиваем координаты и вызываем команду zoom, впоследствии чего перерисовываем фрактал.

actListener() разрешает управление командами. Здесь мы обрабатывает команду от кнопки reset. При срабатывании команды, мы перерисовываем иллюстрацию в изначальное положение.

drawFractal() создает FractalWorker, который высчитывает сочетание RGB цветов и дальше отрисовывает изображение.

doInBackground() Значение сего способа заключается в том, что он делает фоновою операцию, метод вызывается в фоновом режиме и вычисляет все необходимое для отрисовки фрактала.

done() – метод вызывается за это время, когда doInBackground() завершил собственную работу, по завершению вычислений RGB цветов в doInBackground, функция отрисовывает пиксели.

1. Создаем класс BurningShip

getInitialRange() высчитывает,  какая доля плоскости считается самой выгодной  для фрактала.

numIterations()Высчитывает количество итераций для предоставленного фрактала: на протяжении 2000 итераций или же пока же z1 < 4, считает переменную count, которая и считается количеством подходящих итераций.  
Различие у кода фракталов  лишь только в их особенности вычисления

1. Создаем класс Tricorn

getInitialRange() разрешает предопределять какая доля плоскости считается самой многообещающей для фрактала.  
numIterations() Высчитывает количество итераций для предоставленного фрактала: на протяжении 2000 итераций или же пока же z1 < 4, считает переменную count, которая и считается количеством подходящих итераций.  
Отличие различие у кода фракталов лишь только в их особенности вычисления

.

# Вывод:

В ходе выполнения лабораторной работы я приобрел навык написания программ на языке Java, умение создавать ГУИ, создавать там изображения, создавать кнопки, создавать фракталы.

Также, из-за того, что программа использует несколько потоков процессора, фрактал стал отрисовываться намного быстрее и интерфейс перестал зависать.