ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

Ордена Трудового Красного Знамени

федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский Технический Университет Связи и Информатики» (МТУСИ)

Кафедра МКиИТ

Лабораторная работа по технологиям программирования №7

«Многоядерный генератор фракталов»

Выполнил:

Студент 2 курса

Группы БСТ-1602

Абросимов Данил

Вариант №1

Москва, 2018

Оглавление

[1. Цель и задачи. 3](#_Toc507964498)

[1.1. Доработать класс location, реализовав проверку точек на идентичность и реализовав метод кучи. 3](#_Toc507964499)

[1.2. Доработать класс AStarState, реализовав все нужные для работы программы функции.. 3](#_Toc507964500)

[2. Анализ предметной области и выбор инструментария. 3](#_Toc507964501)

[3. Код программы и объяснение функций. 3](#_Toc507964502)

[3.1. Класс Location. 3](#_Toc507964503)

[3.2. Класс AppStarState. 4](#_Toc507964504)

[4. Заключение. 6](#_Toc507964505)

1. Цель и задачи.

Доработать класс FractalExplorer, реализовав проверку точек на идентичность и реализовав метод кучи. Добавить 2 фрактала: Burning Ship, Tricorn.

1. Анализ предметной области и выбор инструментария.

В текущей работе я использовал бесплатно распространяемый пакет разработчика Java Development Kit. Однако, этот пакет не предусматривает интегрированную среду разработки, поэтому для написания непосредственно кода мною была выбрана программа Notepad++, так как она обеспечивает достаточно гибкий интерфейс и предоставляет возможность подсветки синтаксиса языка.

1. Код программы и объяснение функций.
   1. Класс WebCrawler.

mport java.net.\*;

import java.util.\*;

import java.io.\*;

public class WebCrawler {

public static void main(String[] args) {

int depth = 0;

String URL, depth;

if (args.length != 2) {

System.out.println("usage: java Crawler <URL> <depth>");

System.exit(1);

} else {

try {

depth = Integer.parseInt(depth);

} catch (NumberFormatException nfe) {

System.out.println("usage: java Crawler <URL> <depth>");

System.exit(1);

}

}

}

LinkedList<String> getLinks(URL\_DP \_URL){

LinkedList<String> URLs = new LinkedList<String>();

Socket socket;

try {

socket = new Socket(\_URL.getHost(), 80);

}

catch(IOException ex){return null;}

try {

InputStreamReader in = new InputStreamReader(socket.getInputStream());

}

catch(IOException ex){ return null;}

String Path = \_URL.getPath();

String Host = \_URL.getHost();

return URLs ;

}

}

main() проверяет корректность введенных данных, если данные не верны, выводит сообщение об ошибке…

getLinks()

* 1. Класс URL\_DP.

import java.net.\*;

public class URL\_DP {

private int \_Depth;

private String \_URL;

public URL\_DP(String URL, int depth) {

\_Depth = depth;

\_URL = URL;

}

public String getURL() {

return \_URL;

}

public int getDepth() {

return \_Depth;

}

public String ToString() {

String stringDepth = Integer.toString(\_Depth);

return stringDepth + '\t' + \_URL;

}

public String getPath() {

try {

URL url = new URL(\_URL);

return url.getPath();

}

catch (MalformedURLException e) {

return null;

}

}

public String getHost() {

try {

URL url = new URL(\_URL);

return url.getHost();

}

catch (MalformedURLException e) {

return null;

}

}

}

getURL() возвращает текстовый URL.  
getDepth() возвращает глубину.  
getPath() Возвращает текстовый пусть.

getHost() возвращает хост.

ToString() Вывод URL и глубины в одной строке.

1. Заключение.

Проделав всю работу, мы познакомились с java GUI, научились создавать кнопки, выпадающие списки, вставлять картинки, обрабатывать запросы и многое другое. Научились создавать фракталы.