package com.helper;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FilenameFilter;

import java.io.IOException;

import java.io.InputStream;

import java.util.ArrayList;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.filechooser.FileFilter;

import javax.swing.filechooser.FileNameExtensionFilter;

import swinghelper.FileChooserHelper;

public class FileHelper {

public static void main(String[] args) {

StringBuffer sb = getFileContent("D:/114");

System.out.println(sb);

}

public static StringBuffer getFileContent(String filepath) {

InputStream is = null;

int i;

char c;

StringBuffer sb = new StringBuffer();

try {

File f = new File(filepath);

System.out.println(f.getCanonicalPath());

if (!f.exists()) {

System.out.println("File Does NOT exist!!");

return sb;

}

is = new FileInputStream(filepath);

byte[] b = new byte[1024];

ByteArrayOutputStream baos = new ByteArrayOutputStream();

while ((i = is.read(b)) != -1) {

// String s = new String(b);

// sb.append(s.trim());

baos.write(b, 0, i);

}

sb = new StringBuffer(new String(baos.toByteArray()));

} catch (Exception e) {

e.printStackTrace();

} finally {

if (is != null) {

try {

is.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

return sb;

}

public static ArrayList<String[]> parseFile(String fileName) {

ArrayList<String[]> arr = new ArrayList<String[]>();

StringBuffer sb = getFileContent(fileName);

String[] tokens = sb.toString().split("\\|1234\\|");

for (int i = 0; i < tokens.length; i++) {

String string = tokens[i];

String[] keyTweet = string.split("\\|\\|");

arr.add(keyTweet);

}

return arr;

}

public static ArrayList<String[]> parseFile(String fileName, String rowDelim, String colDelim) {

ArrayList<String[]> arr = new ArrayList<String[]>();

StringBuffer sb = getFileContent(fileName);

String[] tokens = sb.toString().split(rowDelim);

for (int i = 0; i < tokens.length; i++) {

String string = tokens[i];

String[] keyTweet = string.split(colDelim);

arr.add(keyTweet);

}

return arr;

}

public static File[] getFileList(String dirPath) {

File f = new File(dirPath);

try {

System.out.println("Canonical Path " + f.getCanonicalPath());

} catch (IOException ex) {

ex.printStackTrace();

}

File[] a = f.listFiles();

if (a != null) {

System.out.println(" Got Files " + a.length);

}

return a;

}

// extn=.txt .jpg

public static File[] getFileList(String dirPath, final String extn) {

File f = new File(dirPath);

try {

System.out.println("Canonical Path " + f.getCanonicalPath());

} catch (IOException ex) {

ex.printStackTrace();

}

FilenameFilter textFilter = new FilenameFilter() {

public boolean accept(File dir, String name) {

String lowercaseName = name.toLowerCase();

if (lowercaseName.endsWith(extn)) {

return true;

} else {

return false;

}

}

};

File[] a = f.listFiles(textFilter);

if (a != null) {

System.out.println(" Got Files " + a.length);

}

return a;

}

public void fileFilter() {

// JFileChooser fileChooser = new JFileChooser();

// System.out.println("1");

//// fileChooser.addChoosableFileFilter(new FileFilter() {

////

//// public String getDescription() {

//// System.out.println("1");

//// return "PDF Documents (\*.pdf)";

//// }

////

//// public boolean accept(File f) {

//// if (f.isDirectory()) {

//// System.out.println("1");

//// return true;

//// } else {

//// System.out.println("1");

//// return f.getName().toLowerCase().endsWith(".pdf");

//// }

//// }

//// });

// fileChooser.addChoosableFileFilter(new FileNameExtensionFilter("Video", "mp4", "avi", "wmv"));

// int response = fileChooser.showOpenDialog(null);

// if (response == JOptionPane.OK\_OPTION) {

//

// }

filePath = FileChooserHelper.selectFile();

}

public String filePath = "";

}