Virtual Key Repository Source Code

package com.pack1;

import java.io.File;

import java.io.IOException;

import java.util.Arrays;

import java.util.Scanner;

public class VirtualKey {

static String DIRECTORY;

File folder\_name;

public VirtualKey() {

DIRECTORY = System.getProperty("user.dir");

folder\_name = new File(DIRECTORY+"/files");

if (!folder\_name.exists())

folder\_name.mkdirs();

System.out.println("DIRECTORY : "+ folder\_name.getAbsolutePath());

}

private static final String WELCOME\_PROMPT =

"\n\*\*\*\*\*\*\* VirtualKey.com \*\*\*\*\*\*\*"+

"\n\*\*\*\*\*\*\* Purna Chandra \*\*\*\*\*\*\*\n";

private static final String MAIN\_MENU\_PROMPT =

"\nMAIN MENU - Select any of the following: \n"+

"1 -> List files in directory\n"+

"2 -> Add, Delete or Search\n"+

"3 -> Exit Program";

private static final String SECONDARY\_MENU\_PROMPT =

" \nSelect any of the following: \n"+

" p -> Add a file\n"+

" q -> Delete a file\n"+

" r -> Search a file\n"+

" s -> GoBack";

void showPrimaryMenu() {

System.out.println(MAIN\_MENU\_PROMPT);

try{

Scanner scanner = new Scanner(System.in);

int option = scanner.nextInt();

switch (option){

case 1 : {

showFiles();

showPrimaryMenu();

}

case 2 : {

showSecondaryMenu();

}

case 3 : {

System.out.println("Thank You sucessfully executed program");

System.exit(0);

}

default: showPrimaryMenu();

}

}

catch (Exception e){

System.out.println("Please enter 1, 2 or 3");

showPrimaryMenu();

}

}

void showSecondaryMenu() {

System.out.println(SECONDARY\_MENU\_PROMPT);

try{

Scanner scanner = new Scanner(System.in);

char[] input = scanner.nextLine().toLowerCase().trim().toCharArray();

char option = input[0];

switch (option){

case 'p' : {

System.out.print("↳ Adding a file...Please Enter a File Name : ");

String filename = scanner.next().trim().toLowerCase();

addFile(filename);

break;

}

case 'q' : {

System.out.print("↳ Deleting a file...Please Enter a File Name : ");

String filename = scanner.next().trim();

deleteFile(filename);

break;

}

case 'r' : {

System.out.print("↳ Searching a file...Please Enter a File Name : ");

String filename = scanner.next().trim();

searchFile(filename);

break;

}

case 's' : {

System.out.println("Going Back to MAIN menu");

showPrimaryMenu();

break;

}

default : System.out.println("Please enter p, q, r or s");

}

showSecondaryMenu();

}

catch (Exception e){

System.out.println("Please enter p, q, r or s");

showSecondaryMenu();

}

}

void showFiles() {

if (folder\_name.list().length==0)

System.out.println("The folder is empty");

else {

String[] list = folder\_name.list();

System.out.println("The files in "+ folder\_name +" are :");

Arrays.sort(list);

for (String str:list) {

System.out.println(str);

}

}

}

void addFile(String filename) throws IOException {

File filepath = new File(folder\_name +"/"+filename);

String[] list = folder\_name.list();

for (String file: list) {

if (filename.equalsIgnoreCase(file)) {

System.out.println("File " + filename + " already exists at " + folder\_name);

return;

}

}

filepath.createNewFile();

System.out.println("File "+filename+" added to "+ folder\_name);

}

void deleteFile(String filename) {

File filepath = new File(folder\_name +"/"+filename);

String[] list = folder\_name.list();

for (String file: list) {

if (filename.equals(file) && filepath.delete()) {

System.out.println("File " + filename + " deleted from " + folder\_name);

return;

}

}

System.out.println("Delete Operation failed. FILE NOT FOUND");

}

void searchFile(String filename) {

String[] list = folder\_name.list();

for (String file: list) {

if (filename.equals(file)) {

System.out.println("FOUND : File " + filename + " exists at " + folder\_name);

return;

}

}

System.out.println("File NOT found (FNF)");

}

public static void main(String[] args) {

System.out.println(WELCOME\_PROMPT);

VirtualKey menu = new VirtualKey();

menu.showPrimaryMenu();

}

}