**Virtual Key for Your Repositories source code:**

package Finalphase1proj;

import java.io.IOException;

import java.util.Scanner;

public class P1P {

static Scanner sc = new Scanner(System.in);

static opera opera = new opera();

public static final String path = "C:\\Users\\hp\\javafsd\\javademos\\Phase1projects\\src\\Phase1project\\demo file created";

public static void main(String[] args) throws IOException {

/\* System.out.println("Hello World!"); \*/

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("\tWelcome to virtual key repository \n");

// System.out.println("\t");

System.out.println("\t Developed by- Rishav Kumar \n");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

optionsSelection1();

}

private static void optionsSelection1() throws IOException {

String[] arr1 = { "1. Retrieving the file names in an ascending order", "2. Business-level operations:",

"3. Close the application" };

int[] arr2 = { 1, 2, 3 };

int slen = arr2.length;

for (int i = 0; i < slen; i++) {

System.out.println(arr1[i]);

// displayed the all the Strings mentioned in the String array

}

System.out.println("\nEnter your choice:\t");

Scanner sc1 = new Scanner(System.in);

int options = sc1.nextInt();

for (int j = 1; j <= slen; j++) {

if (options == j) {

switch (options) {

case 1:

System.out.println();

opera.LAF(P1P.path);

optionsSelection1();

break;

case 2:

String[] arr3 = { "1. Option to add a user specified file to the application",

"2. Option to delete a user specified file from the application",

"3. Option to search a user specified file from the application",

"4. Navigation option to close the current execution context and return to the main context"

};

int[] arr4 = { 1, 2, 3, 4 };

int slen1 = arr4.length;

for (int i = 0; i < slen1; i++) {

System.out.println(arr3[i]);

// display the all the Strings mentioned in the String array

}

System.out.println("Enter the value to perform operation: \n");

Scanner sc2 = new Scanner(System.in);

int options2 = sc2.nextInt();

for (int k = 1; k <= slen1; k++) {

if (options2 == k) {

String file;

String filename;

switch (options2) {

case 1:

System.out.println("Please enter a file name to add : ");

file = sc.nextLine();

filename = file.trim();

opera.createNewFile(P1P.path, filename);

optionsSelection1();

break;

case 2:

System.out.println("Please enter a file name to Delete : ");

file = sc.nextLine();

filename = file.trim();

opera.deleteFile(P1P.path, filename);

optionsSelection1();

break;

case 3:

System.out.println("Please enter a file name to Search : ");

file = sc.nextLine();

filename = file.trim();

opera.searchFile(P1P.path, filename);

optionsSelection1();

break;

case 4:

System.out.println(

"Navigation option to close the current execution context and return to the main context \n");

optionsSelection1();

break;

default:

System.out.println("You have made an invalid choice!");

optionsSelection1();

break;

}

}

}

case 3:

System.out.println("\n Are you sure you want to exit ? ");

System.out.println(" (Y) -> Yes (N) -> No ");

int option = sc.nextLine().toUpperCase().charAt(0);

if (option == 'Y') {

System.out.println("\n");

System.exit(1);

} else if (option == 'N') {

System.out.println("\n");

optionsSelection1();

} else {

System.out.println("\nInvalid Input \nValid Inputs :(Y/N)\n");

optionsSelection1();

}

break;

default:

System.out.println("You have made an invalid choice!");

optionsSelection1();

break;

}

}

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2ndclass\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package Finalphase1proj;

import java.io.File;

import java.io.IOException;

import java.util.Arrays;

import java.util.Set;

import java.util.TreeSet;

import java.util.regex.\*;

public class opera {

public void LAF(String path) {

File directory = new File(path);

String[] files = directory.list();

Set<String> filesList = new TreeSet<String>(Arrays.asList(files));

System.out.println("All Files in location: " + directory.getAbsolutePath() + " are: \n");

for (String file1 : filesList) {

System.out.println(file1);

}

}

public void createNewFile(String path, String fileName) throws IOException {

File newFile = new File(path + File.separator + fileName);

boolean createFile = newFile.createNewFile();

if (createFile) {

System.out.println("\nFile successfully created at location: " + newFile.getAbsolutePath());

} else if (!createFile) {

System.out.println("\nFile Already Exist.. Please try again.");

}

}

public void deleteFile(String path, String fileName) throws IOException {

File directory = new File(path);

File newFile = new File(path + File.separator + fileName);

boolean deleteFile = newFile.delete();

if (deleteFile) {

System.out.println("\nFile deleted Successfully from location:"+ directory.getAbsolutePath());

} else {

System.out.println("\nFile not found...");

}

}

public void searchFile(String path, String fileName) {

File directory = new File(path);

String[] fileList = directory.list();

boolean flag = false;

Pattern pat = Pattern.compile(fileName);

for (String file : fileList) {

Matcher mat = pat.matcher(file);

if (mat.matches()) {

System.out.println("File Found at location: " + directory.getAbsolutePath());

flag = true;

break;

}

}

if (flag == false)

System.out.println("File Not Found.. Please try again.");

}

}