**Virtual Key Source Code**

**KeyBoard.java**

package com.demo.virtualKey;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class KeyBoard {

String headLine = "Welcome to Virtual Key Application";

String developerName;

String path = "./src/main/directory";

public KeyBoard(String developerName) {

this.developerName = developerName;

}

Scanner sc = new Scanner(System.in);

public void developerDashBoard() {

System.out.println(

"====================================================================================================");

System.out.println(headLine);

System.out.println("Developer Name : " + developerName);

}

public void mainMenuDashBoard() {

System.out.println(

"====================================================================================================");

System.out.println("Main Menu");

System.out.println("----------------------");

List<String> mainOpt = new ArrayList<>();

mainOpt.add("1. Show Files");

mainOpt.add("2. Show sub menu");

mainOpt.add("3. Exit");

mainOpt.stream().forEach(System.out::println);

System.out.println(

"====================================================================================================");

}

public void subMenuDashBoard() {

System.out.println(

"====================================================================================================");

System.out.println("Sub Menu");

System.out.println("----------------------");

List<String> subOpt = new ArrayList<>();

subOpt.add("1. Add File");

subOpt.add("2. Delete File");

subOpt.add("3. Search File");

subOpt.add("4. Return main Menu");

subOpt.stream().forEach(x -> System.out.println(x));

System.out.println(

"====================================================================================================");

}

public void showMenu() throws IOException {

int option = 0;

developerDashBoard();

do {

mainMenuDashBoard();

option = sc.nextInt();

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

switch (option) {

case 1: {

showFiles();

break;

}

case 2: {

showSubMenu();

break;

}

case 3: {

System.out.println(

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

break;

}

default:

System.out.println("Invalid option. Please enter again");

}

} while (option != 3);

System.out.println("Thank you for using our application");

}

public void showFiles() {

System.out.println("List of Files");

System.out.println("----------------------");

File f = new File(path);

String[] listOfFiles = f.list();

Arrays.sort(listOfFiles);

for (String s : listOfFiles) {

System.out.println(s);

}

}

public void showSubMenu() throws IOException {

int option = 0;

do {

subMenuDashBoard();

option = sc.nextInt();

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

switch (option) {

case 1: {

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

sc.nextLine();

String fileName = sc.nextLine();

addFile(fileName);

break;

}

case 2: {

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

sc.nextLine();

String fileName = sc.nextLine();

deleteFile(fileName);

break;

}

case 3: {

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

sc.nextLine();

String fileName = sc.nextLine();

searchFile(fileName);

break;

}

case 4: {

break;

}

default:

System.out.println("Invalid option. Please enter again");

}

} while (option != 4);

}

public void addFile(String fileName) throws IOException {

File f = new File(path, fileName);

if (fileName == null || fileName == "") {

System.out.println("File name can't be null....");

} else {

if (f.exists() == false) {

f.createNewFile();

System.out.println("File added Succefully with name : " + fileName);

} else {

System.out.println("File already exists..with name : " + fileName);

}

}

}

public void deleteFile(String fileName) {

File f = new File(path, fileName);

boolean file = f.isFile();

boolean status = f.delete();

if (fileName.equals(null) || fileName == "") {

System.out.println("File name can't be empty..");

} else {

if (file == true) {

if (status == true) {

System.out.println("File deleted successfully with name : " + fileName);

}

} else {

System.out.println("Please enter correct file name..");

}

}

}

public void searchFile(String fileName) {

File f = new File(path, fileName);

boolean status = f.exists();

if (fileName.equals(null) || fileName == "") {

System.out.println("File name can't be empty..");

} else {

if (status == true) {

System.out.println("You are searching for a file named : " + fileName);

System.out.println("Found : " + fileName);

} else {

System.out.println("You are searching for a file named : " + fileName);

System.out.println("File not found");

}

}

}

}

**App.java**

**package** com.demo.virtualKey;

**import** java.io.IOException;

**public** **class** App {

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

KeyBoard k = **new** KeyBoard("Suneel");

**try** {

k.showMenu();

} **catch** (Exception e) {

System.***out***.println("Please enter numbers only.... ");

}

}

}

**pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.demo</groupId>

<artifactId>VirtualKey</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>VirtualKey</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.target>1.8</maven.compiler.target>

<maven.compiler.source>1.8</maven.compiler.source>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>