**Developer details**

**Ajay Augustine**

*Github* **:** [**git-intel (AjayAugustine) / Projects · GitHub**](https://github.com/git-intel?tab=projects)

**GitHub Links**

* *Project:*

[**Java-LockedMe**](https://github.com/users/git-intel/projects/4)*(https://github.com/users/git-intel/projects/4)*

*Linked repositories:*

[**Java\_LockedMe\_Repo**](https://github.com/git-intel/Java_LockedMe_Repo)

*(https://github.com/git-intel/Java\_LockedMe\_Repo)*

**Core concepts**

1. Exception handling in java
2. Collection
3. File handling in java

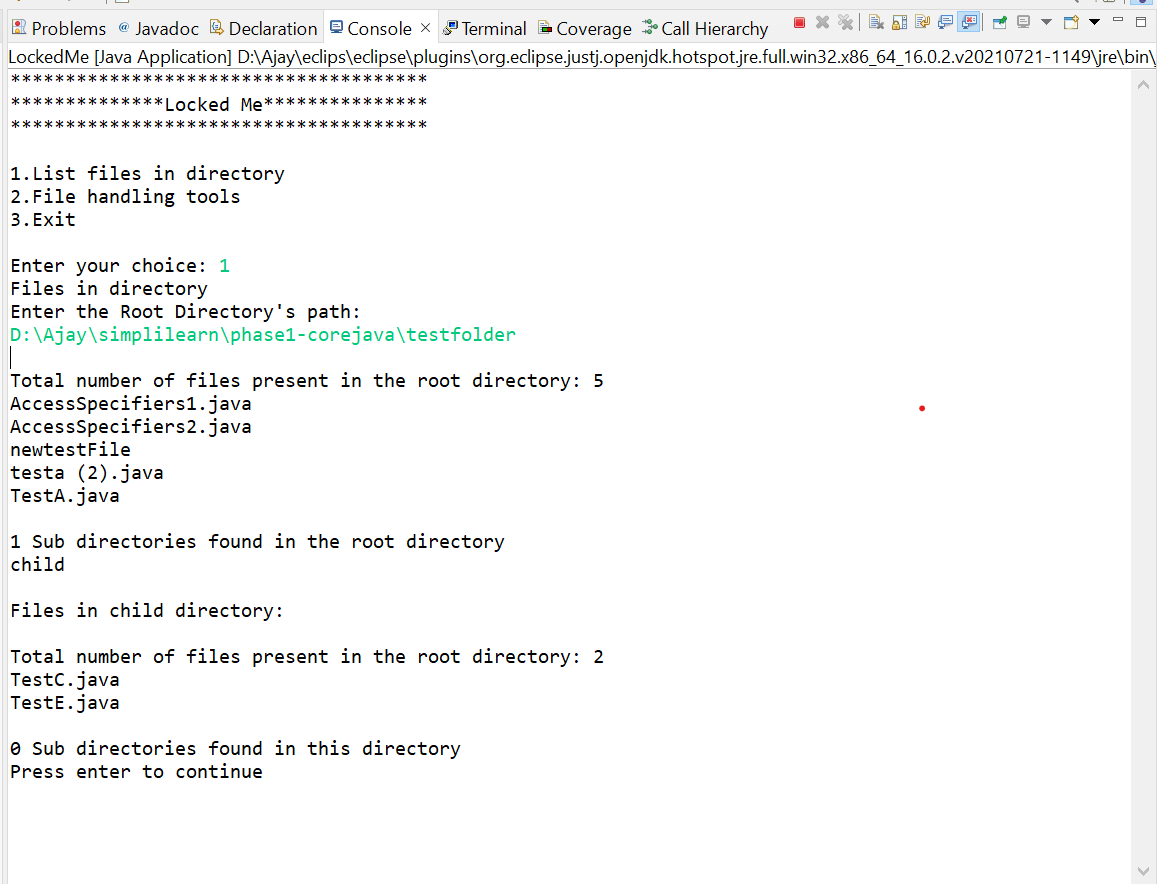
**Sprint Plan**

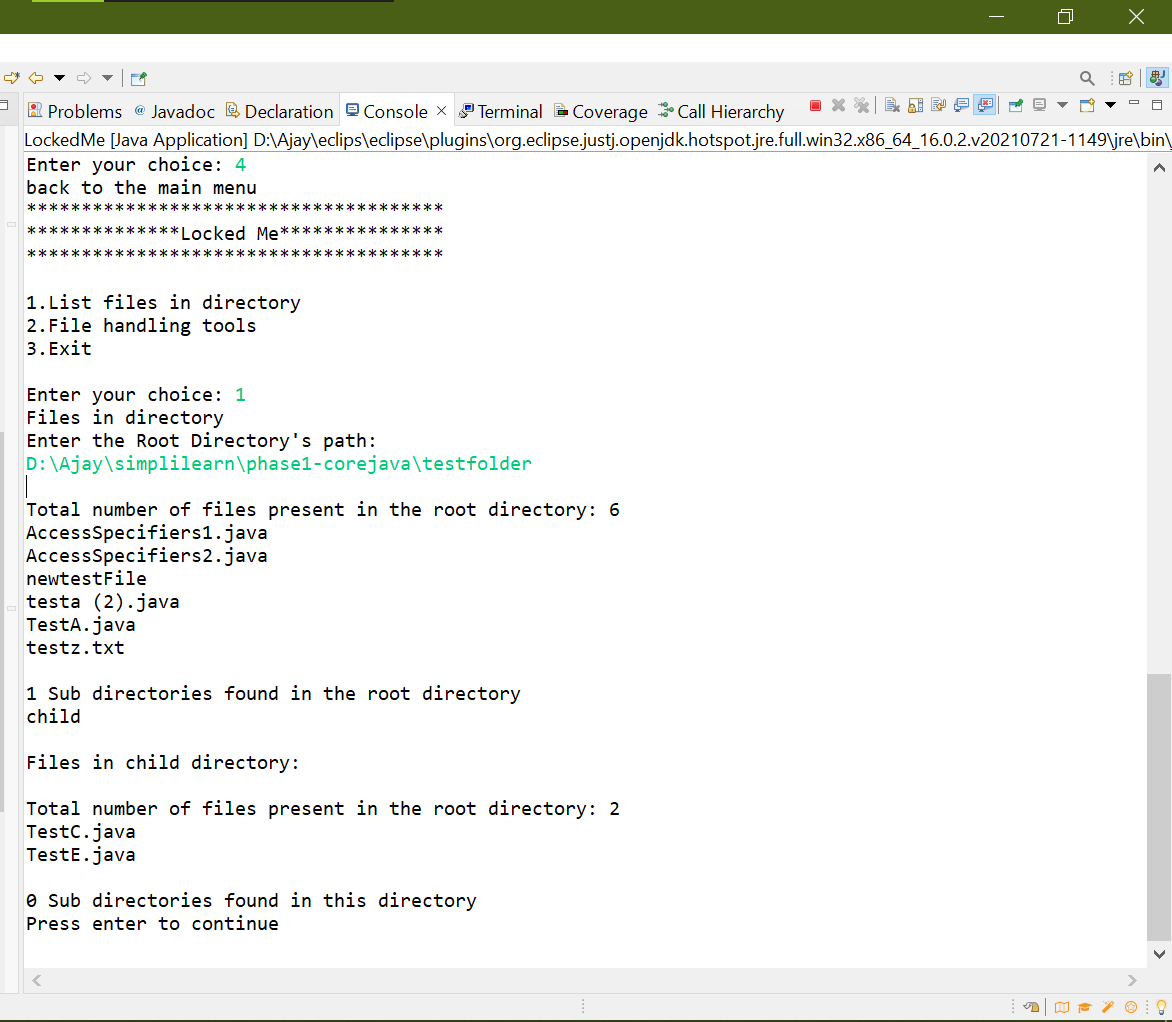
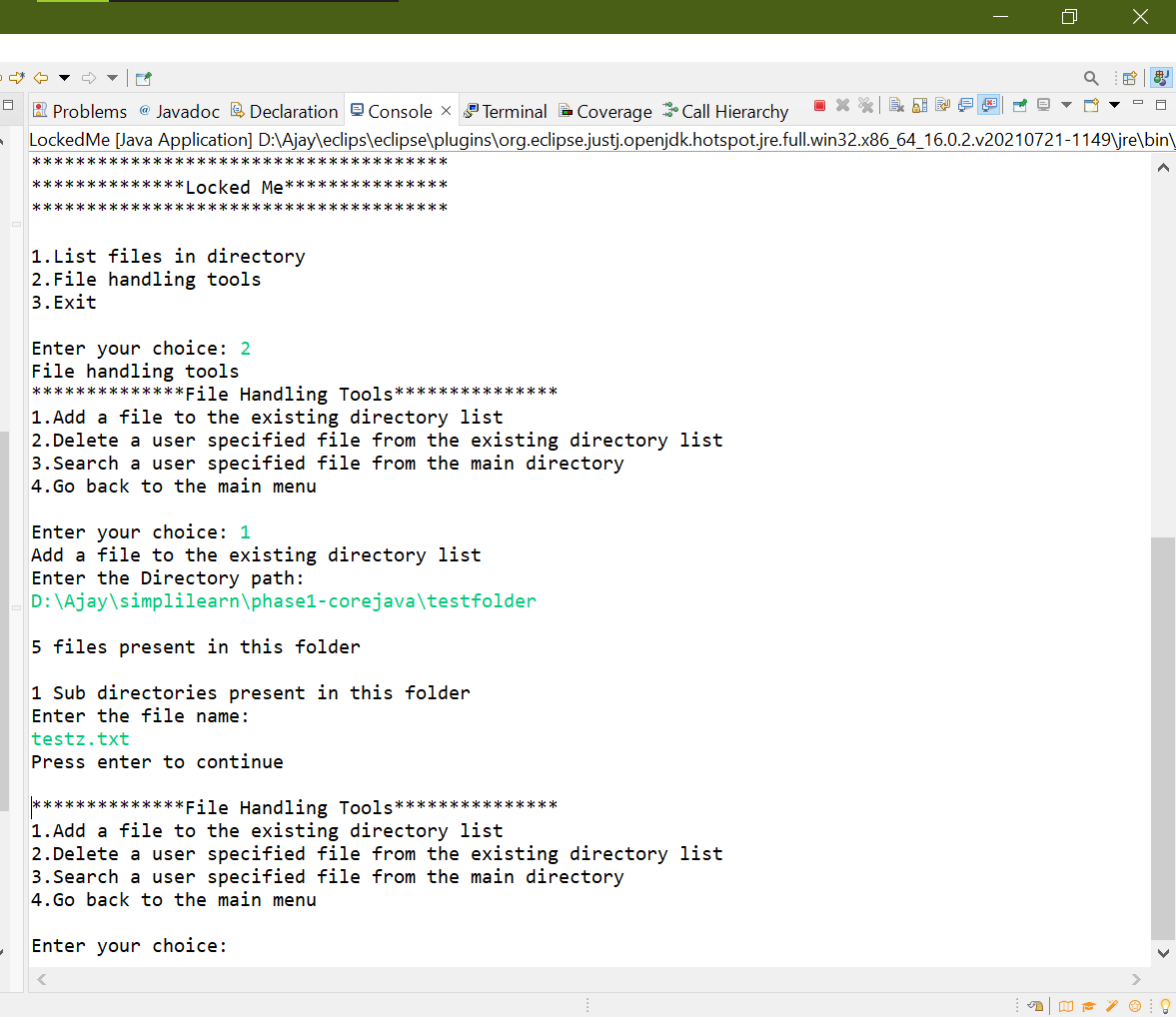


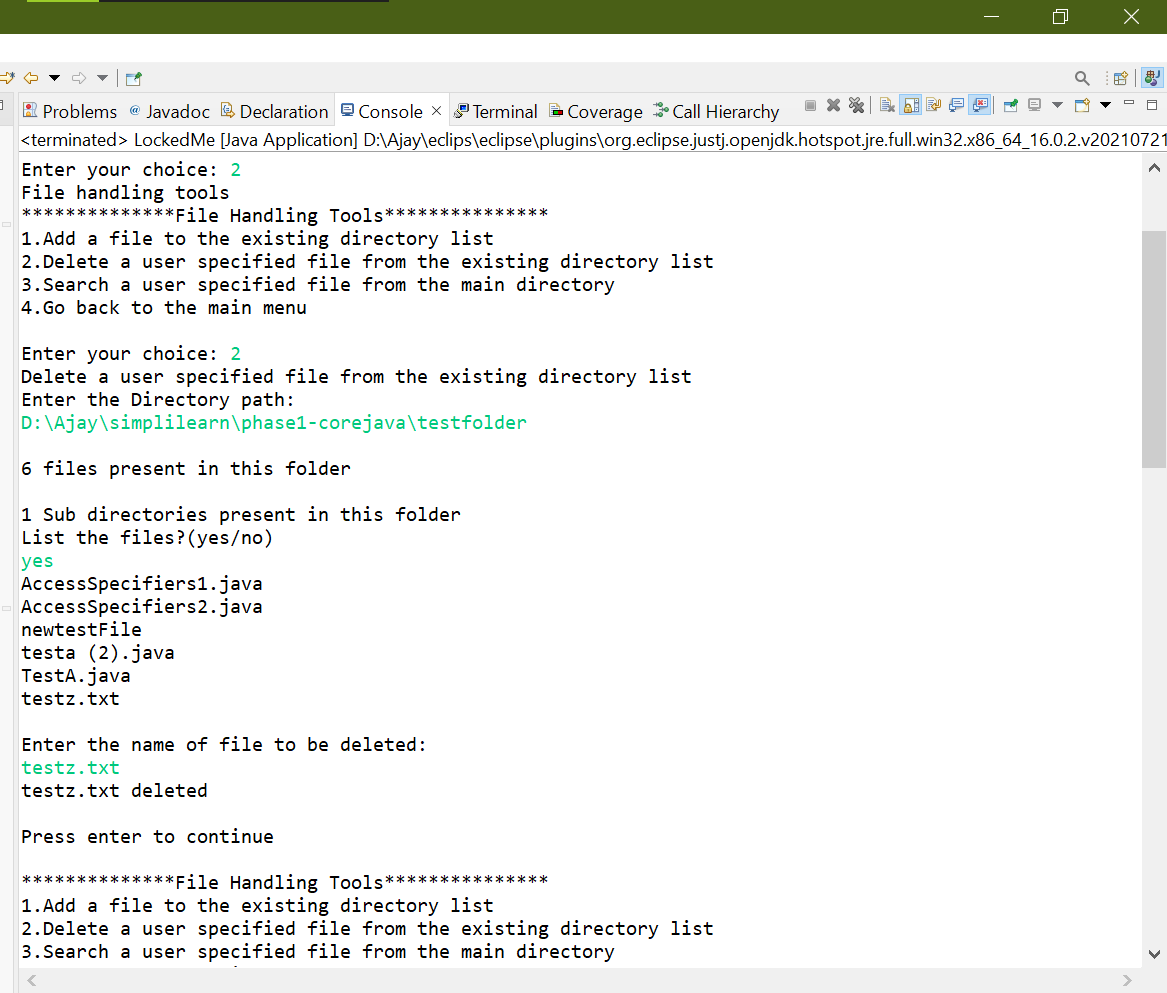


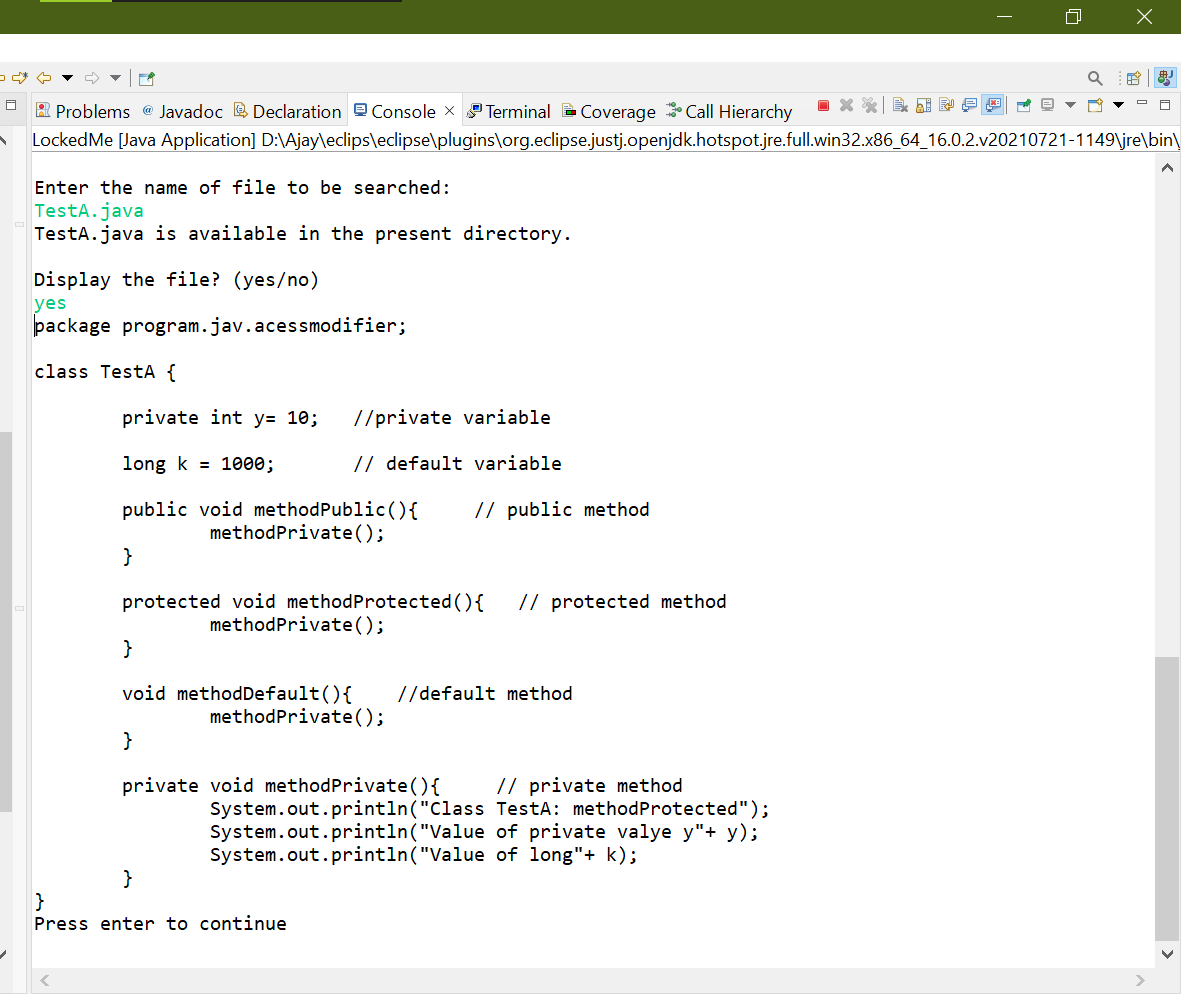
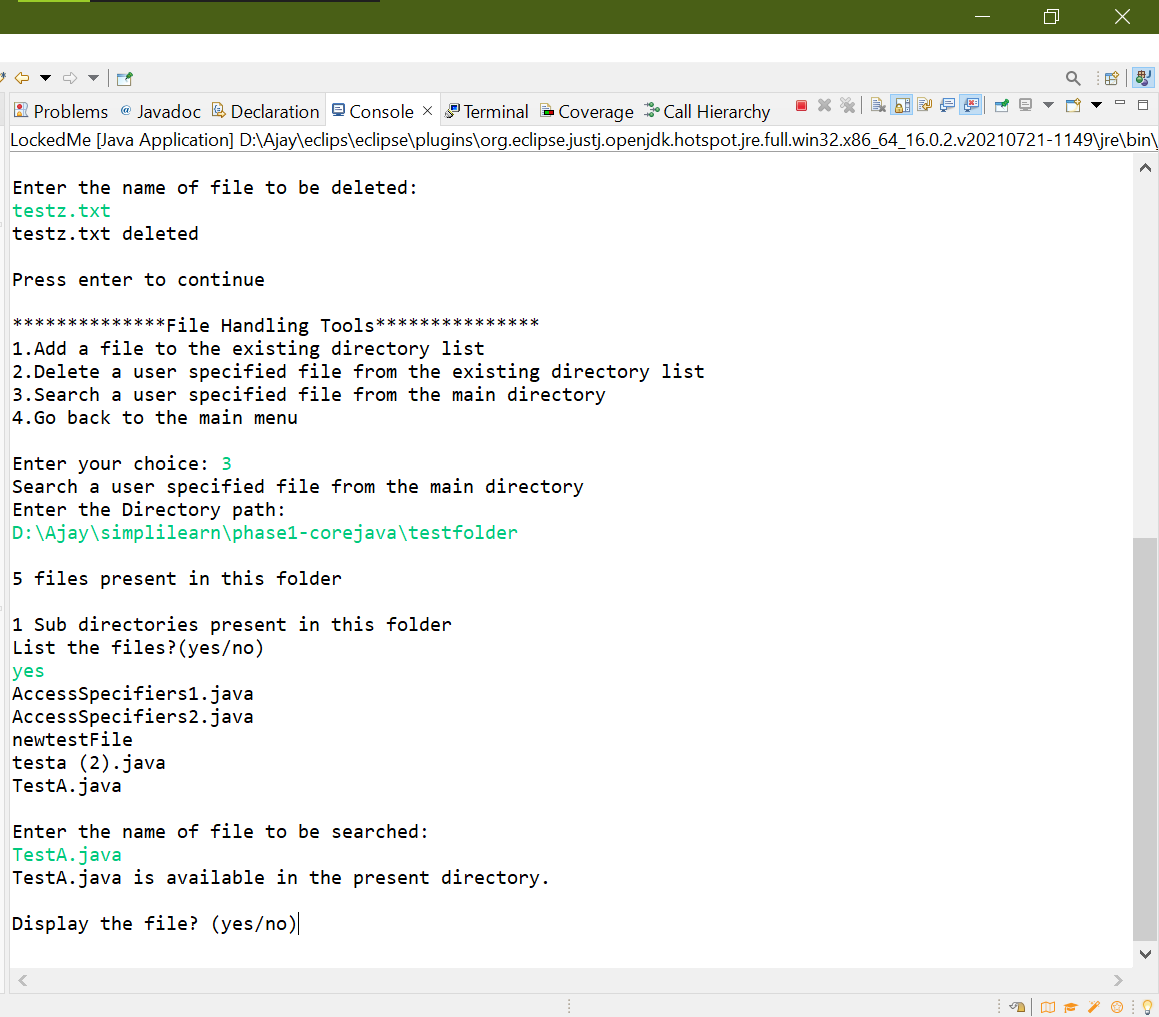
**Output Screens**

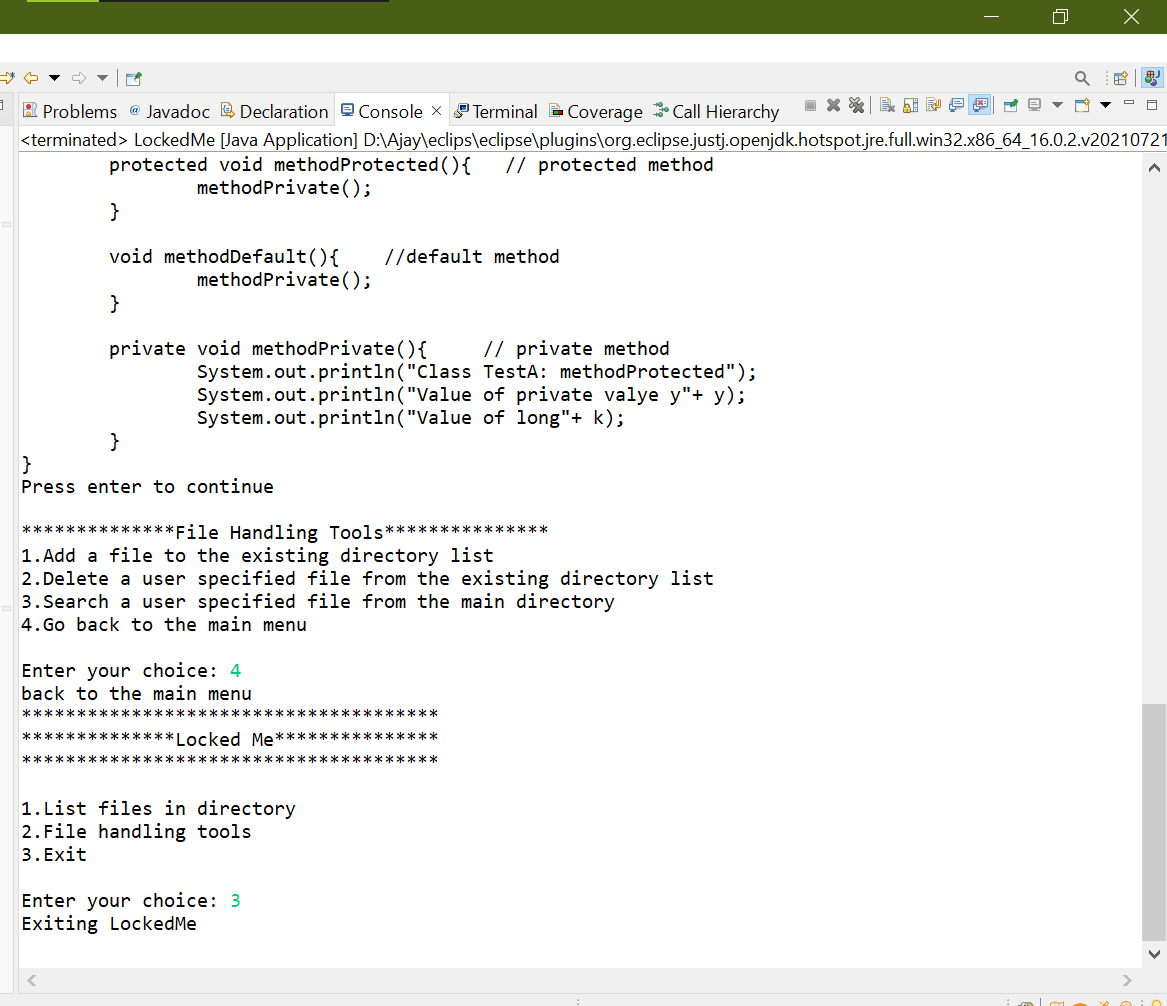
1. List Files in directory



1. Add file to existing directory list
2. Delete a user specified file from the existing directory list



1. Search a user specified file from the main directory.
2. Exit LockedMe



**Source Code**

*package app.lockedme;*

*import java.io.File;*

*import java.io.IOException;*

*import java.util.ArrayList;*

*import java.util.Collections;*

*import java.util.Scanner;*

*public class LockedMe {*

*static boolean optionsAvaiable;*

*static boolean fileoptionsAvaiable;*

*public LockedMe() {*

*}*

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

*optionsAvaiable = true;*

*fileoptionsAvaiable = true;*

*showUserOptions();*

*}*

*public static void showUserOptions() {*

*Scanner userIn = new Scanner(System.in);*

*try {*

*if (optionsAvaiable == false) {*

*System.out.println("Press enter to continue");*

*try {*

*String keyPress = userIn.nextLine();*

*optionsAvaiable = (keyPress != null) ? true : false;*

*} catch (Exception e) {*

*System.out.println(e);*

*}*

*}*

*if (optionsAvaiable == true) {*

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

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

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

*System.out.println("1.List files in directory");*

*System.out.println("2.File handling tools");*

*System.out.println("3.Exit" + "\n");*

*System.out.print("Enter your choice: ");*

*try {*

*int choice;*

*choice = userIn.nextInt();*

*LockedMe lm = new LockedMe();*

*switch (choice) {*

*case 1:*

*System.out.println("Files in directory");*

*optionsAvaiable = false;*

*lm.listFilesInRoot();*

*break;*

*case 2:*

*System.out.println("File handling tools");*

*optionsAvaiable = false;*

*fileoptionsAvaiable = true;*

*LockedMe.showFileHandlingOptions();*

*break;*

*case 3:*

*System.out.println("Exiting LockedMe");*

*optionsAvaiable = false;*

*System.exit(0);*

*break;*

*default:*

*System.out.println("Please select the given options");*

*optionsAvaiable = true;*

*LockedMe.showUserOptions();*

*}*

*} catch (Exception e) {*

*System.out.println(e);*

*}*

*}*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*userIn.close();*

*}*

*}*

*public void listFilesInRoot() {*

*Scanner scanner = new Scanner(System.in);*

*System.out.println("Enter the Root Directory's path: ");*

*try {*

*String directoryPath = scanner.nextLine();*

*File folder = new File(directoryPath);*

*if (folder.isDirectory()) {*

*File[] fileList = folder.listFiles(File::isFile);*

*File[] dirList = folder.listFiles(File::isDirectory);*

*System.out.println("\nTotal number of files present in the root directory: " + fileList.length);*

*for (File file : fileList) {*

*System.out.println(file.getName());*

*}*

*System.out.println("\n" + dirList.length + " Sub directories found in the root directory ");*

*for (File file : dirList) {*

*System.out.println(file.getName());*

*System.out.println("\nFiles in " + file.getName() + " directory: ");*

*LockedMe dr = new LockedMe();*

*dr.listAllFilesInDir(file.getAbsolutePath());*

*}*

*}*

*showUserOptions();*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*scanner.close();*

*}*

*}*

*public void listAllFilesInDir(String directoryPath) {*

*try {*

*File folder = new File(directoryPath);*

*if (folder.isDirectory()) {*

*File[] fileList = folder.listFiles(File::isFile);*

*File[] dirList = folder.listFiles(File::isDirectory);*

*// Arrays.sort(fileList);*

*System.out.println("\nTotal number of files present in the root directory: " + fileList.length);*

*ArrayList<String> al = new ArrayList<String>();*

*/\**

*\* Collections.sort method is sorting the elements of ArrayList in ascending*

*\* order.*

*\*/*

*Collections.sort(al);*

*for (File file : fileList) {*

*al.add(file.getName());*

*}*

*Collections.sort(al);*

*for (String fileName : al) {*

*System.out.println(fileName);*

*}*

*System.out.println("\n" + dirList.length + " Sub directories found in this directory ");*

*for (File file : dirList) {*

*System.out.println(file.getName());*

*System.out.println("\nFiles in " + file.getName() + " directory: ");*

*LockedMe dr = new LockedMe();*

*dr.listAllFilesInDir(file.getAbsolutePath());*

*}*

*}*

*} catch (Exception e) {*

*System.out.println(e);*

*}*

*}*

*public static void showFileHandlingOptions() {*

*Scanner userIn = new Scanner(System.in);*

*try {*

*if (fileoptionsAvaiable == false) {*

*System.out.println("Press enter to continue");*

*try {*

*String keyPress = userIn.nextLine();*

*fileoptionsAvaiable = (keyPress != null) ? true : false;*

*} catch (Exception e) {*

*System.out.println(e);*

*}*

*}*

*if (fileoptionsAvaiable == true) {*

*System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*File Handling Tools\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");*

*System.out.println("1.Add a file to the existing directory list");*

*System.out.println("2.Delete a user specified file from the existing directory list");*

*System.out.println("3.Search a user specified file from the main directory");*

*System.out.println("4.Go back to the main menu" + "\n");*

*System.out.print("Enter your choice: ");*

*try {*

*int choice;*

*choice = userIn.nextInt();*

*LockedMe lm = new LockedMe();*

*switch (choice) {*

*case 1:*

*System.out.println("Add a file to the existing directory list");*

*fileoptionsAvaiable = false;*

*lm.addFileToDir();*

*break;*

*case 2:*

*System.out.println("Delete a user specified file from the existing directory list");*

*fileoptionsAvaiable = false;*

*lm.deleteFileFromDir();*

*break;*

*case 3:*

*System.out.println("Search a user specified file from the main directory");*

*fileoptionsAvaiable = false;*

*lm.searchFileFromRootDir();*

*break;*

*case 4:*

*System.out.println("back to the main menu");*

*optionsAvaiable = true;*

*LockedMe.showUserOptions();*

*default:*

*System.out.println("Please select the given options");*

*fileoptionsAvaiable = true;*

*LockedMe.showFileHandlingOptions();*

*}*

*} catch (Exception e) {*

*System.out.println(e);*

*}*

*}*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*userIn.close();*

*}*

*}*

*public void addFileToDir() {*

*Scanner scanner = new Scanner(System.in);*

*System.out.println("Enter the Directory path: ");*

*try {*

*String directoryPath = scanner.nextLine();*

*File folder = new File(directoryPath);*

*if (folder.isDirectory()) {*

*File[] fileList = folder.listFiles(File::isFile);*

*File[] dirList = folder.listFiles(File::isDirectory);*

*System.out.println("\n" + fileList.length + " files present in this folder");*

*System.out.println("\n" + dirList.length + " Sub directories present in this folder");*

*System.out.println("Enter the file name: ");*

*String filename = scanner.nextLine();*

*// Using file pointer creating the file.*

*File newfile = new File(folder, filename);*

*if (!newfile.exists()) {*

*newfile.createNewFile();*

*}*

*} else {*

*System.out.println("\n" + "Directory not found");*

*}*

*showFileHandlingOptions();*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*scanner.close();*

*}*

*}*

*public void deleteFileFromDir() {*

*Scanner scanner = new Scanner(System.in);*

*System.out.println("Enter the Directory path: ");*

*try {*

*String directoryPath = scanner.nextLine();*

*File folder = new File(directoryPath);*

*if (folder.isDirectory()) {*

*File[] fileList = folder.listFiles(File::isFile);*

*File[] dirList = folder.listFiles(File::isDirectory);*

*System.out.println("\n" + fileList.length + " files present in this folder");*

*System.out.println("\n" + dirList.length + " Sub directories present in this folder");*

*System.out.println("List the files?(yes/no)");*

*String a = scanner.nextLine();*

*if (a.equals("yes") || a.equals("y")) {*

*for (File file : fileList) {*

*System.out.println(file.getName());*

*}*

*}*

*System.out.println("\nEnter the name of file to be deleted: ");*

*String filename = scanner.nextLine();*

*File todeletefile = new File(folder, filename);*

*if (todeletefile.exists() && filename.equals(todeletefile.getName())) { // not case sensitive*

*if (todeletefile.delete()) {*

*System.out.println(todeletefile.getName() + " deleted\n");*

*} else {*

*System.out.println("delete operation failed");*

*}*

*} else {*

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

*}*

*} else {*

*System.out.println("\n" + "Directory not found");*

*}*

*showFileHandlingOptions();*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*scanner.close();*

*}*

*}*

*public void searchFileFromRootDir() {*

*Scanner scanner = new Scanner(System.in);*

*System.out.println("Enter the Directory path: ");*

*try {*

*String directoryPath = scanner.nextLine();*

*File folder = new File(directoryPath);*

*if (folder.isDirectory()) {*

*File[] fileList = folder.listFiles(File::isFile);*

*File[] dirList = folder.listFiles(File::isDirectory);*

*System.out.println("\n" + fileList.length + " files present in this folder");*

*System.out.println("\n" + dirList.length + " Sub directories present in this folder");*

*System.out.println("List the files?(yes/no)");*

*String a = scanner.nextLine();*

*if (a.equals("yes") || a.equals("y")) {*

*for (File file : fileList) {*

*System.out.println(file.getName());*

*}*

*}*

*System.out.println("\nEnter the name of file to be searched: ");*

*String filename = scanner.nextLine();*

*File tosearchfile = new File(folder, filename);*

*if (tosearchfile.exists() && filename.equals(tosearchfile.getName())) { // not case sensitive*

*System.out.println(tosearchfile.getName() + " is available in the present directory.\n");*

*System.out.println("Display the file? (yes/no)");*

*String ans = scanner.nextLine();*

*if (ans.equals("yes") || ans.equals("y")) {*

*Scanner sc = new Scanner(tosearchfile);*

*try {*

*while (sc.hasNextLine()) {*

*System.out.println(sc.nextLine());*

*}*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*sc.close();*

*}*

*}*

*} else {*

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

*}*

*} else {*

*System.out.println("\n" + "Directory not found");*

*}*

*showFileHandlingOptions();*

*} catch (Exception e) {*

*System.out.println(e);*

*} finally {*

*scanner.close();*

*}*

*}*

*}*