```
import java.io.BufferedReader;
import java.io.File;
import java.io.InputStreamReader;
import java.text.DecimalFormat;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import java.util.concurrent.Executors;
import java.util.concurrent.ThreadPoolExecutor;

public class product
{
 static int chkFlag = 0;
 public void showSortedFiles() {
// Try-catch Block for file detail
      try
      {
         System.out.println("Enter folder path for file details");
         Scanner a1 = new Scanner(System.in);
         String folderPath =  a1.next();
         File dir = new File(folderPath);

         File[] files = dir.listFiles();

         Arrays.sort(files, (f1, f2) -> {
           if (f1.isDirectory() && !f2.isDirectory()) {
           return -1;
```

```
            }
            else if (!f1.isDirectory() && f2.isDirectory()) {
            return 1;
            }
            else {
            return f1.compareTo(f2);
            }});

          for (File file : files) {
            if (!file.isHidden()) {
            if (file.isDirectory()) {
                  System.out.println("DIR \t" + file.getName());
               }
             else {
                  System.out.println("FILE\t" + file.getName());
               }}}}
      catch (Exception ex1) {
      }}

 // Try-catch Block for file detail
public void FileExit(String name, File file) {
      try {
         File[] list = file.listFiles();
         if (list != null) {
         for (File fil : list) {
               if (fil.isDirectory()) {
                  FileExit(name, fil);
        }
         else if (name.equalsIgnoreCase(fil.getName())) {
```

```
                System.out.println(fil.getParentFile());
                chkFlag = 1;
              }}}}
      catch (Exception ex2)
      {}}

// Try-catch Block for  Search file
 public void searchFile() {
    try{
         product ff = new  product();
         Scanner scan = new Scanner(System.in);
         System.out.println("Enter the file to be searched.. ");
         String name = scan.next();
         System.out.println("Enter the directory where to search ");
         String directory = scan.next();
        ff.FileExit(name, new File(directory));
         if (chkFlag == 1){
            System.out.println("File is present:");
            chkFlag = 0;}
         else{
            System.out.println("File is not present:");
         }}
      catch (Exception ex3)
      {}}

 // Try-catch Block for show/navigate file
 public void showFile() {
 try {
         System.out.println("Enter the path for files to search ");
```

```
        Scanner a1 = new Scanner(System.in);
        String folderPath = a1.next();
        File folder = new File(folderPath);
        System.out.println("Enter the file to search for files");

      if (folder.isDirectory()) {
          String[] listOfFiles = folder.list();
        if (listOfFiles.length < 1)
            System.out.println(" file not found");
        else
            System.out.println("List of Files & Folder");
        for (String fileName : listOfFiles) {
              System.out.println(fileName);
          }}
        else
          System.out.println("There is no Folder @ given path :" + folderPath);
      }
catch (Exception ex4)
      {}}

 // Create aNew File
   public void newFile() {
     String strPath = "", strName = "";
 // Try-catch Block
try  {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter the file name:");  // Creating BufferedReadered object
        strName = br.readLine();
       System.out.println("Enter the file path:");  // Reading File name
```

```
            strPath = br.readLine(); // Reading File Path
            File file1 = new File(strPath + "" + strName + ".txt"); // Creating File Object
            file1.createNewFile();}
 catch (Exception ex5)
       {} }
public void deleteFile() {
String strPath = "", strName = "";
try {
  // Creating File Object
            File file1 = new File(strPath + strName + ".txt");
            if (file1.delete()) {
               System.out.println("Deleted the file: " + file1.getName());
            }
            else {
               System.out.println("Failed to delete the file.");
            }}
       catch (Exception ex6){
            System.out.println(ex6.getStackTrace());
       }}

 public static void main(String[] args) throws Exception {
       product fileobj = new  product ();
        Scanner scan = new Scanner(System.in);

       System.out.println(" <<<<<<<<<<<<<****HII!!..WELCOME TO JAVA
PROGRAM****>>>>>>>>>>>>>");
       System.out.println("Developer: Rutuja Sontakke");

       while (true) {
```

```
  System.out.println("You Are in Main menu:");
  System.out.println("1: Show Files");
  System.out.println("2: File Option Menu");
  System.out.println("3: Quit");

  System.out.println("   Enter your choice : ");
int choice = scan.nextInt();// accept user input
switch (choice) {
  case 1:
        System.out.println("   1: SHOW FILES ");
        fileobj.showSortedFiles();
        System.out.println("  Return Back to Main Menu");
  break;

  case 2:
        System.out.println(" 2:FILE MENU OPTION");
        System.out.println("1: File Creation");
        System.out.println("2: File  Deletion");
        System.out.println("3: File Search");
        System.out.println("4: Return to Main Menu");

       int choice2 = scan.nextInt();
       switch (choice2) {
       case 1:
             System.out.println(" Entered for Creating file");
             fileobj.newFile();
       break;
       case 2:
             System.out.println(" Entered for Deleting file");
```

```
            fileobj.deleteFile();
        break;
        case 3:
            System.out.println(" Entered for Search file");
            fileobj.searchFile();
       break;
       case 4:
            System.out.println(" Return Back to Main Menu:");
       break; }


 break;
 case 3:
        System.out.println(" you quit ");
        System.exit(0);
break;
default:
        System.out.println("INVALID CHOICE..!!!");
}}}}
```