**File Manager 1.0**

***by Md.Mazher***

“ File Manager 1.0 ” Application consists of two menus. The Main menu and other one is Sub menu. This application basically accepts input from the user, process it on the basis of choice that the user made and generate out.

**Main Menu**

For the main menu, I declare a string array called “ arr ”. Which basically prints 3 options on the user's screen, so that user can figure out which option is referred to which operation. To print string array, enhanced for loop is used

*For (String val: arr){*

*System.out.prinln(val);*

*}*

After that, I declare another int variable called "choice", which stores user input and passes it to switch case. I use the switch case for better functionality, which takes the user's input and handles it on the basis of choices that the user's choice and finally give you results. It consists of 3 cases :

* File in Ascending order
* Sub Menu (Add, delete and search)
* Close application

*Switch(choice){*

*Case 1:*

*fileInAscendingOrder();*

*mainMenu();*

*break;*

*Case 2:*

*subMenu();*

*break;*

*Case 3:*

*closeApp();*

*break;*

*}*

If user press 1, It will call “ **fileInAscendingOrder** ” method. This returns the list of files in ascending order. To do so, I first declare the string " currentDirectory " variable. Used to store the present location of the directory.

*String currentDirectory = System.getProperty(“user.dir”)*

I used File class in Java, represent the file or directory path name. After that I created the File object and passed the variable currentDirectory.

*File newFolder = new File(currentDirectory);*

Next, declare string array " fileNames". fileNames will store the list of all available files in the current directory, converting the object into a list and storing it in the fileNames array.

*String []fileNames = new String[50];*

*fileNames = newFolder.list();*

Now we have files, to sort it. I have used traditional for loop to sort file in ascending. Another enhanced for loop to print files names.

*int size = fileNames.length;*

*for(int i = 0; i<size-1; i++) {*

*for (int j = i+1; j<fileNames.length; j++) {*

*if(fileNames[i].compareTo(fileNames[j])>0) {*

*String temp = fileNames[i];*

*fileNames[i] = fileNames[j];*

*fileNames[j] = temp;*

*}*

*}*

*}*

*for(String name : fileNames) {*

*System.out.println("File : " + name);*

*}*

If user press 2, Then it will takes user to “ **Sub Menu** ”.

If press 3, then it will call “ **closeApplication** ”method. Which closes the application.

*public static void closeApp() {*

*System.out.println("\tClosing Application... \n\t Thank you!");*

*}*

**Sub Menu**

In the submenu, I declare a string called "arr1". Which mainly prints 4 options on the user screen, so the user can understand which option refers to which operation. For printing a string array, enhanced for the loop is used.

*For (String val: arr1){*

*System.out.prinln(val);*

*}*

After that, I declare another variable called "choice2", which stores the user input and passes it to Switch case. I use the switch case for better functionality, which takes the user input and processes it on the basis of choice that the user choice and finally give you the output. It consists of 4 cases:

* Add file in existing directory
* Delete file from existing directory
* Search for file
* Return to main menu

*Switch(choice2){*

*Case 1:*

*addFile();*

*subMenu();*

*break;*

*Case 2:*

*deleteFile();*

*subMenu();*

*break;*

*Case 3:*

*searchFile();*

*subMenu();*

*break;*

*Case 4:*

*mainMenu();*

*break;*

*}*

When you press 1, the " **addFile** " method is called. It will add a file in an existing directory. For this I declare string “ **strFileNames** ” variable which stores user input file name. Next, I created a file object by passing the createDirectory and strFileNames variable and checking whether the file is already present with the same name or not. If present, then the user needs to put different name. If not, under try-catch block creates the file.

*String strFileNames = input.next();*

*File theFile = new File(currentDirectory, strFileNames);*

*if(theFile.isFile()) {*

*System.out.println("File with this name already exist, try with another name");*

*}*

*else {*

*try {*

*theFile.createNewFile();*

*System.out.println("file is created successfully");*

*}catch (IOException e) {*

*System.out.println(e.getMessage());*

*}*

*}*

By pressing 2, "**deleteFile** " method called. It will delete the particular file from current directory. To achieve this, I declare another string variable called "strFileNames1". Which store user input. Then again, I created file object by passing **createDirectory** and **strFileNames1** variable and using if-else statement check if file is already present or not. If present, then delete the file. If not, then print message, delete operation not possible.

*public static void deleteFile() {*

*String strFileNames1 = input.next();*

*File theFile = new File(currentDirectory, strFileNames1);*

*if(theFile.isFile()) {*

*theFile.delete();*

*System.out.println("file is deleted successfully");*

*}*

*else {*

*System.out.println("------Delete operation not possible, file does not exist------");*

*}*

*}*

*}*

If press 3, “ **searchFile** ” method called. It will search for a particular file in directory. For this I declare another variable string variable called “ **strFileNames2** ”. Which store user input. Then again, I created file object by passing **createDirectory** variable. Then converting object into list of file and storing it into string array called “ **fileNames** ”. Also declare Boolean variable called “ **flag** ” and initialize with false. Now use for loop for iteration over an array and using if-else statement, whether the file is present or not. If file present, set the **flag** to true and break from for loop. Now check **flag**, If it is true, print file found. If it is false, print file not found.

*public static void searchFile()* *{*

*String strFileNames2 = input.next();*

*File newFolder = new File(currentDirectory);*

*String []fileNames = new String[50];*

*fileNames = newFolder.list();*

*boolean flag=false;*

*for(String val:fileNames) {*

*if(val.equals(strFileNames2)) {*

*flag=true;*

*break;*

*}*

*}*

*if(flag) {*

*System.out.println("file is found: "+strFileNames2);*

*}*

*else {*

*System.out.println("file is not found: "+strFileNames2);*

*}*

*}*

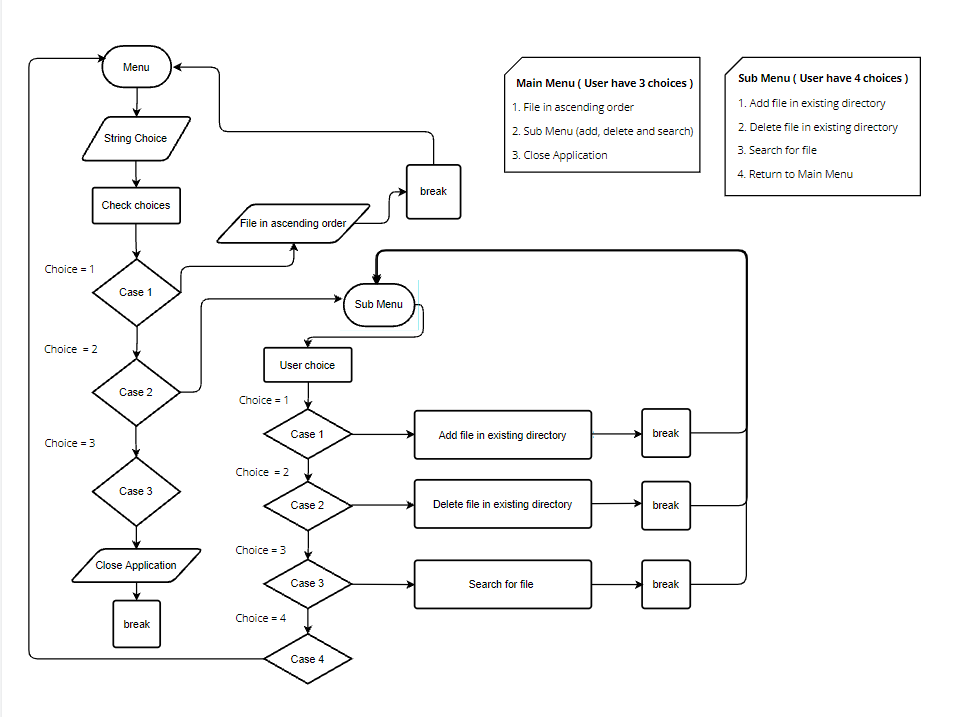
If press 4, then it take back to you to the main menu.

*case 4:*

*mainMenu();*

*break;*

**Flow Chart**

****

**Git Link**

*https://github.com/mdmazher01/Phase-1\_FinalAssignment.git*