

# Java FSD with Angular: Course End Project 1

## Source Code:

### FolderHandling.java:

```
package com.project.resources.classes;

import java.io.File;
import java.util.Arrays;
import java.util.Comparator;
import java.util.List;

public class FoldersHandling {
    private File directory;

    public FoldersHandling(String path) {
        this.directory = new File(path);
    }

    public List<File> listAllFiles() throws NullPointerException {
        List<File> files = Arrays.asList(this.directory.listFiles());
        files.sort(new Filesort());
        return files;
    }

    public File getDirectory() {
        return this.directory;
    }
}

class Filesort implements Comparator<File> {
    @Override
    public int compare(File o1, File o2) {
        return o1.getName().compareTo(o2.getName());
    }
}
```

### FileOperations.java:

```
package com.project.resources.classes;

import java.io.File;
import java.io.IOException;;

public class FileOperations {
    private static FoldersHandling foldersHandling;

    public FileOperations(FoldersHandling foldersHandling) {
        FileOperations.foldersHandling = foldersHandling;
    }

    public void createFile(String filename) {
        File directory = foldersHandling.getDirectory();
        File newFile = new File(directory.toString() + "\\\" + filename);
        if (directory.exists() && directory.isDirectory()) {
            if (!this.searchFile(filename)) {
                try {
                    if (newFile.createNewFile()) {
```

```

        System.out.println(filename + " file
created");
    } else {
        System.out.println(filename + " not
created try again");
    }
} catch (IOException e) {
    System.out.println("Error happens please try
again");
}
} else {
    System.out.println(filename+" already exist");
}
} else {
    System.out.println("Folder not Found");
}
}

public void deleteFile(String filename) {
    File directory = foldersHandling.getDirectory();
    File newFile = new File(directory.toString() + "\\\" + filename);
    if (directory.exists() && directory.isDirectory()) {
        if (this.searchFile(filename)) {
            if (newFile.delete()) {
                System.out.println(filename + " Deleted
Successfully");
            } else {
                System.out.println("New file not created
created try again");
            }
        } else {
            System.out.println("File does not exist");
        }
    } else {
        System.out.println("Folder not Found");
    }
};

public boolean searchFile(String filename) {
    File directory = foldersHandling.getDirectory();
    File file = new File(directory.toString() + "\\\" + filename);
    if (foldersHandling.listAllFiles().contains(file))
        return true;
    return false;
}
}

```

## Main.java:

```

package com.project.main;

import java.io.File;
import java.util.InputMismatchException;
import java.util.List;
import java.util.Scanner;

import com.project.resources.classes.FileOperations;
import com.project.resources.classes.FoldersHandling;

public class Main {

```

```

static Scanner input = new Scanner(System.in);
static final String PATH = "D:\\Java FSD\\Course End Project
1\\CourseEndProject_1\\src\\com\\project\\resources\\data";
static FoldersHandling foldersHandling = new FoldersHandling(PATH);

public static void main(String[] args) {
    developerInfo();
    String op;
    boolean flag = true;
    while (flag) {
        System.out.println("-----
-----");
        System.out.println("| Main Menu
|");
        System.out.println("-----
-----");
        System.out.println(
            "1. Fetch all files\n2. Do operations(Create,
Delete, Search)\nOther option leads to exit\nEnter your option:");
        op = input.next();
        input.nextLine();
        switch (op) {
            case "1":
                fileList();
                break;
            case "2":
                operationOptions();
                break;
            default:
                System.out.println("-----Thank You for using the
Application-----");
                System.exit(0);
        }
    }
}

private static void fileList() {
    try {
        List<File> files = foldersHandling.listAllFiles();
        if (files.size() > 0) {
            System.out
                .println("-----Files in " +
foldersHandling.getDirectory().getName() + " directory are-----");
            for (File file : files)
                System.out.println(file.getName());
        } else {
            System.out.println("Folder is empty");
        }
    } catch (NullPointerException e) {
        System.out.print("Folder Not found");
    }
}

private static void operationOptions() {
    boolean flag = true;
    String filename, op;
    FileOperations fileOperations = new FileOperations(foldersHandling);
    while (flag) {
        System.out.println("-----
-----");
    }
}

```

```

        System.out.println("| Operations
|");
        System.out.println("-----");
        System.out.println(
            "1. Create a file\n2. Delete a file\n3. Search
a file\n4. Back to main menu\nEnter Your option(Any other digits to exit):");
        try {
            op = input.next();
            input.nextLine();
            switch (op) {
                case "1":
                    System.out.print("Enter the file name to
create: ");

                    filename = input.nextLine();
                    fileOperations.createFile(filename);
                    break;
                case "2":
                    System.out.print("Enter the file name to
delete: ");

                    filename = input.nextLine();
                    fileOperations.deleteFile(filename);
                    break;
                case "3":
                    System.out.print("Enter the file name to
search: ");

                    filename = input.nextLine();
                    if (fileOperations.searchFile(filename))
                        System.out.println(filename + " found");
                    else
                        System.out.println(filename + " not
found");

                    break;
                case "4":
                    flag = false;
                    break;
                default:
                    System.out.println("-----Thank You for using
the Application-----");

                    System.exit(0);
            }
        } catch (InputMismatchException ime) {
            System.out.println("Please enter the valid options");
        }
    }

    private static void developerInfo() {
        System.out.println("-----");
        System.out.println("| Application name
|");
        System.out.println("| Developer: Manoj M
|");
    }
}

```