

```

import java.io.IOException;
import java.nio.file.*;
import java.nio.file.attribute.BasicFileAttributes;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Main {

    private static final String SPACE = " ";
    private static Map<Path, FileInfo> directoryFileInfoMap = new HashMap<>();

    static class FileInfo {
        int count = 0;
        long size = 0;
    }

    public static void main(String[] args) {
        //User Input
        Scanner scanner = new Scanner(System.in);
        System.out.printf("Please enter the file path\n");
        String input = scanner.nextLine();
        Path startPath = Paths.get(input);

        //Path Validation

        //Recursive File Display
        try {
            Files.walkFileTree(startPath, new SimpleFileVisitor<Path>() {
                private int level = 0;

                @Override
                public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes attrs) {
                    if (!dir.equals(startPath)) {
                        FileInfo info = new FileInfo();
                        directoryFileInfoMap.put(dir, info);
                    }
                    printSPACE(level);
                    System.out.println("> " + dir.getFileName());
                    level++;
                    return FileVisitResult.CONTINUE;
                }
            });
        }
    }

```

```

@Override
public FileVisitResult visitFile(Path file, BasicFileAttributes attrs) {
    FileInfo info = directoryFileInfoMap.get(file.getParent());
    if (info != null) {
        info.count++;
        info.size += attrs.size();
    }
    printSPACE(level);
    System.out.println("- " + file.getFileName());
    return FileVisitResult.CONTINUE;
}

@Override
public FileVisitResult postVisitDirectory(Path dir, IOException exc) {
    level--;
    if (level > 0) {
        FileInfo info = directoryFileInfoMap.get(dir);
        if (info != null) {
            printSPACE(level);
            System.out.println(">> " + dir.getFileName() + " - Files: " + info.count + ", Size: " + info.size
+ " bytes");
        }
    }
    return FileVisitResult.CONTINUE;
}

private void printSPACE(int level) {
    System.out.print(SPACE.repeat(level));
}
});
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```