

```

package edu.citadel.csci603.util;

import java.io.File;
import java.nio.file.*;
import java.nio.file.attribute.BasicFileAttributes;
import java.io.IOException;
import static java.nio.file.FileVisitResult.*;

/**
 * Utility class that prints the directory structure to standard output
 * showing the composition of nested files and subdirectories.
 */
public class PrintDirectoryStructureVisitor extends SimpleFileVisitor<Path> {

    private int nestingLevel = 0;

    /**
     * Prints the structure for the file whose path name is given in arg[0].
     */
    public static void main(String[] args) throws IOException {
        if (args.length != 1) {
            printUsage();
            System.exit(-1);
        }

        String pathName = args[0];
        Path startingDir = Paths.get(pathName);

        PrintDirectoryStructureVisitor vis = new PrintDirectoryStructureVisitor();
        Files.walkFileTree(startingDir, vis);
    }

    /**
     * Prints the current file with the current nestingLevel value and continues walk.
     * Doesn't currently use the BasicFileAttributes arg, might consider removing
     param
     * and removing override.
     */
    @Override
    public FileVisitResult visitFile(Path file, BasicFileAttributes bfAttrs) {
        System.out.println(getIndentString(nestingLevel) + "- " +
            file.getFileName());
        return CONTINUE;
    }

    /**
     * Before visiting the dir, increments the nestingLevel and prints the dir name.
     * Doesn't currently use the BasicFileAttributes arg, might consider removing
     param
     * and removing override.
     */
    @Override
    public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes bfAttrs) {
        System.out.println(getIndentString(nestingLevel++) + "+ " +
            dir.getFileName());
        return CONTINUE;
    }

    /**
     * After visiting the dir, decrements the nestingLevel and continues the walk
     */
    @Override
    public FileVisitResult postVisitDirectory(Path dir, IOException e)
        throws IOException

```

```
{
    if(e == null) {
        nestingLevel--;
    } else throw e;

    return CONTINUE;
}

// Use inherited implementation for method visitFileFailed()

private static String getIndentString(int nestingLevel) {
    StringBuilder s = new StringBuilder();
    for (int i = 0; i < nestingLevel; i++)
        s.append(" ");
    return s.toString();
}

private static void printUsage() {
    System.out.println("Usage: edu.citadel.csis603.util.<path>");
    System.out.println("    where <path> is the path of a file or directory");
    System.out.println();
}
}
```