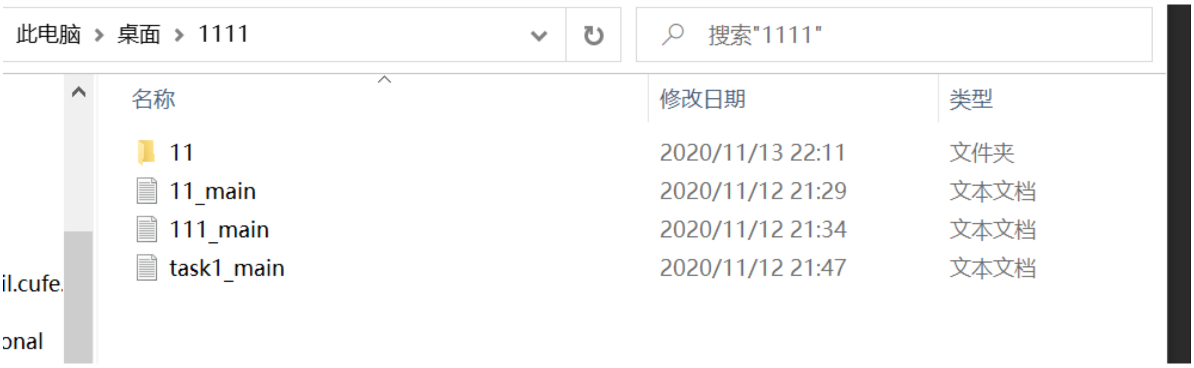
java第四次作业

```java
import java.util.*;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.security.MessageDigest;

public class hello2 {
    public static void main(String[] args){
        dfs_show("C:\\Users\\Lenovo\\Desktop\\1111");
    }

    public static byte[] SHA1Checksum(ArrayList<File> paths) throws Exception {

        byte[] buffer = new byte[1024];


        MessageDigest complete = MessageDigest.getInstance("SHA-1");

        int numRead = 0;
        for (File p : paths) {
            FileInputStream is = new FileInputStream(p);

            do {

                numRead = is.read(buffer);

                if (numRead > 0) {

                    complete.update(buffer, 0, numRead);
                }


            } while (numRead != -1);


            is.close();
        }

        return complete.digest();
    }


    public static String showhash(ArrayList<File> paths) {
        StringBuilder result = new StringBuilder();
        try {
            byte[] sha1 = SHA1Checksum(paths);
            for (byte i : sha1) {
                result.append(Integer.toString(i, 16));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return result.toString();
    }
    public static ArrayList<File> dfs(String path, ArrayList<File> paths)
    {
//对文件夹进行深度优先遍历，返回文件夹中所有文件路径
        File dir = new File(path);
//返回该文件夹中文件的抽象路径名数组
        File[] fs = dir.listFiles();
        if(fs != null) {
//采用 ArrayList 对 listFiles()的结果进行排序
            for(File f : fs) {
                if (f.isFile()) {
                    paths.add(f);
                }
//路径为文件夹，则继续对文件夹进行遍历
                if (f.isDirectory()) {
                    paths = dfs(path + File.separator + f.getName(),paths);
                }
            }
            paths.sort(Comparator.naturalOrder());
        }
        return paths;
    }
    public static void dfs_show(String path) {
//对文件深度遍历并求出各文件与文件夹的 Hash 值
        File dir = new File(path);
        ArrayList<File> paths0 = new ArrayList<>();ArrayList<File> paths =
dfs(path, paths0);
        System.out.println("directory " + dir.getName()+ " Hash 值：  " +
                showhash(paths));
//遍历文件夹中文件并计算各文件或子文件夹 hash 值
        File[] fs = dir.listFiles();
        if(fs != null) {
            for(File f : fs) {
//路径为文件
                if (f.isFile()) {
                    paths = new ArrayList<>();
                    paths.add(f);
                    System.out.println("file " + f.getName()+ " Hash 值：  "
                            + showhash(paths));
                }
//路径为文件夹
                if (f.isDirectory()) {
                    dfs_show(path + File.separator + f.getName());
                }
            }
        }
//加入语句区分文件夹层次
        System.out.println("directory " + dir.getName()+ " end");
    }
}
```

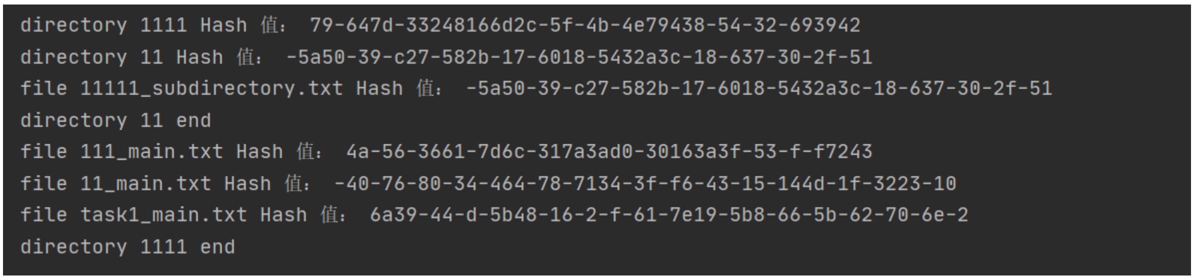
运行结果

- 原文件夹状态



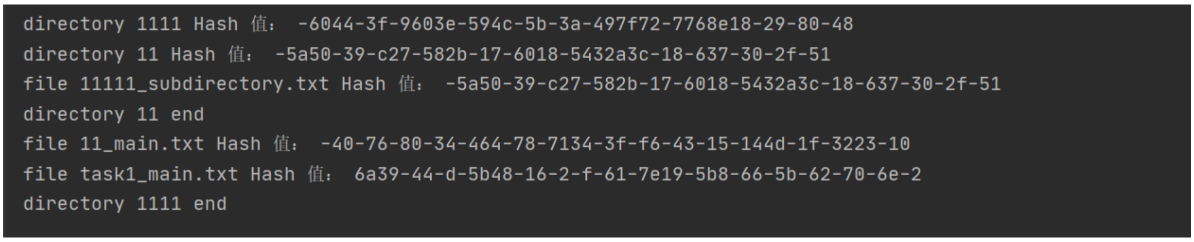
名称	修改日期	类型
11	2020/11/13 22:11	文件夹
11_main	2020/11/12 21:29	文本文档
111_main	2020/11/12 21:34	文本文档
task1_main	2020/11/12 21:47	文本文档

- 哈希值计算结果



```
directory 1111 Hash 值： 79-647d-33248166d2c-5f-4b-4e79438-54-32-693942
directory 11 Hash 值： -5a50-39-c27-582b-17-6018-5432a3c-18-637-30-2f-51
file 11111_subdirectory.txt Hash 值： -5a50-39-c27-582b-17-6018-5432a3c-18-637-30-2f-51
directory 11 end
file 111_main.txt Hash 值： 4a-56-3661-7d6c-317a3ad0-30163a3f-53-f-f7243
file 11_main.txt Hash 值： -40-76-80-34-464-78-7134-3f-f6-43-15-144d-1f-3223-10
file task1_main.txt Hash 值： 6a39-44-d-5b48-16-2-f-61-7e19-5b8-66-5b-62-70-6e-2
directory 1111 end
```

- 更改一个文件-发现相关文件夹的哈希值都发生了改变。逻辑上是正确的。



```
directory 1111 Hash 值： -6044-3f-9603e-594c-5b-3a-497f72-7768e18-29-80-48
directory 11 Hash 值： -5a50-39-c27-582b-17-6018-5432a3c-18-637-30-2f-51
file 11111_subdirectory.txt Hash 值： -5a50-39-c27-582b-17-6018-5432a3c-18-637-30-2f-51
directory 11 end
file 11_main.txt Hash 值： -40-76-80-34-464-78-7134-3f-f6-43-15-144d-1f-3223-10
file task1_main.txt Hash 值： 6a39-44-d-5b48-16-2-f-61-7e19-5b8-66-5b-62-70-6e-2
directory 1111 end
```