

JAVA 编程进阶上机报告



学 院 _____ 智能与计算学部 _____
专 业 _____ 软件工程 _____
班 级 _____ 2 班 _____
学 号 _____ 3018216095 _____
姓 名 _____ 郭紫珊 _____

Homework4 File Copy

1 实验目的

把一个文件从源文件夹复制到目的文件夹。

2 实验过程

首先我考虑的比较简单。只考虑到了一种情况，那就是源文件夹里面都是文件，不包括目录。

所以自然而然地就是打开源文件夹，循环将源文件夹中的文件复制到目的文件夹中。

其中需要使用 `sourceFolder.list()` 这个方法，将文件夹中的所有文件的文件名储存在字符串数组中。然后 `for (String file : files)` 循环将文件复制到目标文件夹中。

```
public static void copy(File sourceFolder, File destinationFolder) throws
IOException{
    String[] files = sourceFolder.list();
    for (String file : files){
        File srcFile = new File(sourceFolder, file);
        File destFile = new File(destinationFolder, file);
        Files.copy(srcFile.toPath(), destFile.toPath());
    }
}
```

其中 `Files.copy(source, target, options)` 是一个java自带的方法，比流读写的方法要简单。

源码部分：

```
public static Path copy(Path source, Path target, CopyOption... options)
throws IOException
{
    FileSystemProvider provider = provider(source);
    if (provider(target) == provider) {
        // same provider
        provider.copy(source, target, options);
    } else {
        // different providers
        CopyMoveHelper.copyToForeignTarget(source, target, options);
    }
    return target;
}
```

从源码可以认识到，首先要创造两个file对象，一个是源文件，一个是复制之后的文件，获取path后传入函数。option可以不传值。

而且这个方法要求，1) 复制之后的文件target原本不存在，否则就会报错。2) 目标文件的路径必须要有，源代码方法中没有创建dir的方法。

那么我们使用 `Files.copy(source, target, options)` 方法，我们需要在刚刚的 `copy()` 方法中考虑到目标路径不存在的情况，所以要加上：

```
if (!destinationFolder.exists()) {
    destinationFolder.mkdir();
}
```

然后我考虑到，源文件夹里面可能包含有目录和文件，目录里面又会有目录或者文件。那么需要复制的不仅仅是文件，还有目录也要完整地复制过去。既然目录是一个树状结构，我们可以使用递归的方法。先判断源路径是文件还是目录。如果是文件，那么直接复制；如果是目录，则遍历目录里的文件，把这个目录里面的文件一个个复制过去。递归直到源文件的最后一层复制结束。

修改之后的代码如下：

```
public static void copy(File sourceFolder, File destinationFolder) throws
IOException{
    if (sourceFolder.isDirectory()){
        if (!destinationFolder.exists()) {
            destinationFolder.mkdir();
        }
        String[] files = sourceFolder.list();
        for (String file : files){
            File srcFile = new File(sourceFolder, file);
            File destFile = new File(destinationFolder, file);
            copy(srcFile, destFile);
        }
    }
    else {
        Files.copy(sourceFolder.toPath(), destinationFolder.toPath());
        System.out.println("将 " + sourceFolder.getName()+"复制到"+
destinationFolder.getName());
    }
}
```

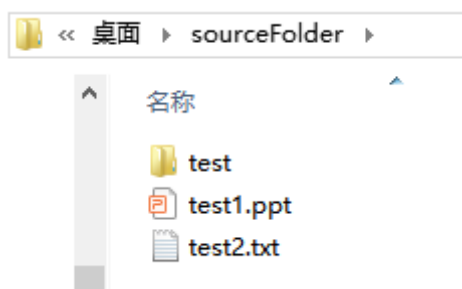
3 测试结果

main函数如下：

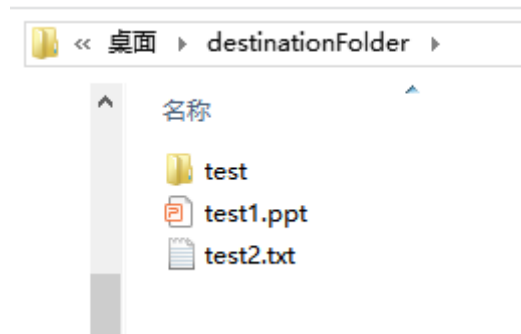
```
public static void main(String[] args) throws IOException {

    File sourceFolder = new File("C:/Users/lenovo/Desktop/sourceFolder");
    File destinationFolder = new
File("C:/Users/lenovo/Desktop/destinationFolder");
    copy(sourceFolder, destinationFolder);
}
```

执行main函数之前：



执行main函数之后：



控制台输出：

