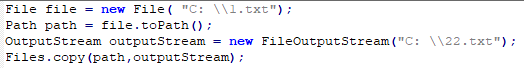
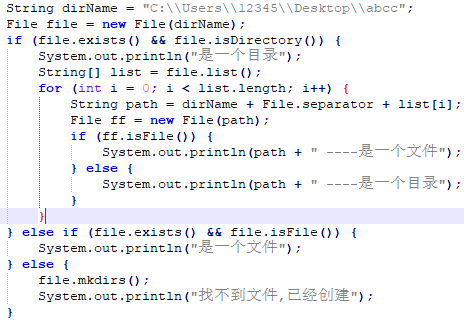
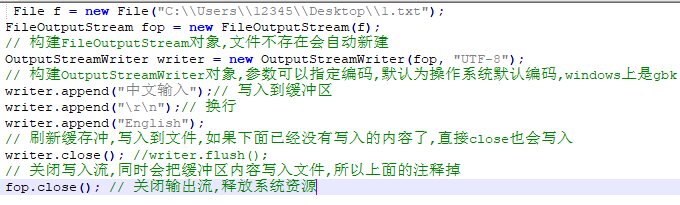
* 1. **文件拷贝：java.nio.Files**



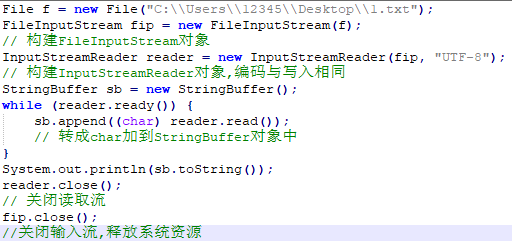
* 1. **文件操作 目录新建、删除、遍历**



**1.3文件读： 使用OutPutStream.writer**



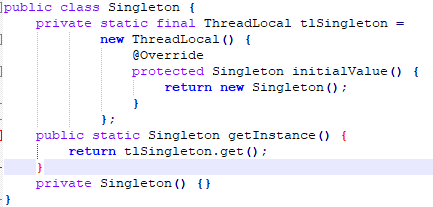
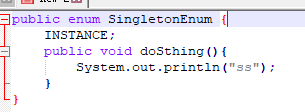
**1.4写到文件 使用InputStreamReader.read**



**2.1 线程池Demo / 或者使用Excutors.newFixedThreadPool**



**3.1 单例模式 ：使用枚举最好。ThreadLocal线程安全/传统方式加Synchronize**



**3.2 策略模式**

策略容器R接入声明 策略接口A并在构造器赋值A，从而按需抽取A中方法包装。具体调用时传入A的实现类（向上造型）**。**

**3.3 观察者模式**

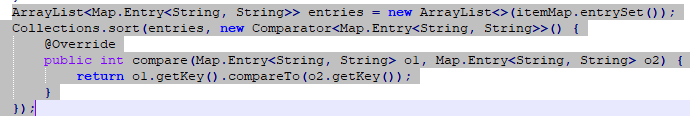
观察者Subject,被观察者Observer; Subject接口定义观察者add,remove,收到通知后的动作A，其实现类 使用List容器存放被观察者Observer. 并在add,remove方法中使用该List增删。通知后的动作A实现方式就是遍历，逐个通知到Observer.

**3.4 代理模式**

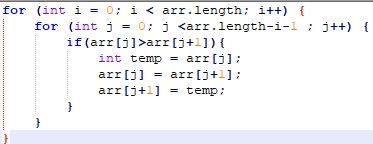
代理类A、被代理类B实现同一接口C，具备了一模一样的行为。不过在代理类A接入声明被代理类B、实现方法中使用被实现类B对象调用相应方法，达到代理调用B的方法的目的。

———————————————————————————————————————

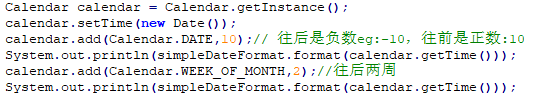
**4.1 Map 元素自定义排序，包装成List ，再使用Collections.sort(list,comparator)**



**4.2 冒泡排序**



5.1 日期、时间



5.2 集合查找 int Collections.binarySearch(list,item) 返回索引

5.3 交集listA.retain(listB) \ 并集setA.addAll(setB) \ 差集 listA.removeAll(listB)

5.4 数字精度处理 BIgDecimal \ NumberFormat

BigDecimal –add(BigDecimal) 、subtract(BigDecimal)、multiply(BigDecimal)、divide(BigDecimal)、intValue()等

setScale(1)表示保留一位小数，默认用四舍五入方式   
setScale(1,BigDecimal.ROUND\_DOWN)直接删除多余的小数位，如2.35会变成2.3   
setScale(1,BigDecimal.ROUND\_UP)进位处理，2.35变成2.4   
setScale(1,BigDecimal.ROUND\_HALF\_UP)四舍五入，2.35变成2.4

setScaler(1,BigDecimal.ROUND\_HALF\_DOWN)四舍五入，2.35变成2.3，如果是5则向下舍

