

# 《Java 技术》实验报告

实验二：

2020 年 5 月 25 日

学院	软件学院	专业班级	软件工程 18-4 办	姓名	刘铭源		成绩											
课程名称	Java 技术	实验项目名称	货物进销管理系统			指导教师	薛峰											
教师评语	<div>教师签名：_____ 年 月 日</div>																	
<div><b>一、实验目的</b><ol style="list-style-type: none"><li>1. 掌握 Java 中文件的读写操作。</li><li>2. 学会使用 Java 提供的实用类（Vector，ArrayList）来完成特定的功能。</li><li>3. 掌握字符串类（String，StringBuffer）的使用。</li><li>4. 掌握用面向对象的方法分析和解决复杂问题。</li></ol><b>二、实验原理</b><p>编写一个 Inventory.java 完成以下功能：</p><ol style="list-style-type: none"><li>1. 程序首先打开并读取 Inventory.txt 中记录的所有库存记录，然后读取 Transactions.txt，处理这个文件中包含的事务，记录发货记录到 Shipping.txt，并记录错误信息到 Errors.txt 中。最后更新库存到另外一个文件 NewInventory.txt 中。</li><li>2. 文件 Inventory.txt 和 NewInventory.txt 的每行包含一个存货记录，没条记录包含下面一些字段息，这些字段之间用一个 tab 分开（见后面的文件格式）：</li></ol><table border="1"><thead><tr><th>字段</th><th>格式和含义</th></tr></thead><tbody><tr><td>Item number</td><td>字符串型，货物编号</td></tr><tr><td>Quantity</td><td>整型，货物数量</td></tr><tr><td>Supplier</td><td>字符串型，供应商编号</td></tr><tr><td>Description</td><td>字符串型，货物描述</td></tr></tbody></table><ol style="list-style-type: none"><li>3. 字段 Items 按照从小到大的顺序写入文件的。注意 Item 号不必连续，如 Item 号为 752 的后面可能是 800。</li><li>4. 文件 Transactions.txt 包含几个不同的处理记录（每行一条记录）。每条记录前面以一个大写字母开头，表示这条记录是什么类型的事务。在不同的大写字母后面是不同的信息格式。所有的字段也是以 tab 键分开的（见 Transactions.txt 文件格式）。</li><li>5. 以'O' 开头（Order 的首字母）的事务表示这是一个发货订单，即某一种货物应该发给特定的客户。Item number 和 Quantity 的格式如上面表格定义。Custom 编号和上面的 Supplier 编号一致。处理一条定单记录（以'O' 开头的事务）意味着从减少库存记录中相应货物的数量（减少的数量=发货单中的数量），记录发货信息到 Shipping.txt 中。注意：Inventory.txt 中的 quantity 不应该小于 0，如果对于某一种货物，库存的数量小于发货单的数量的话，系统应该停止处理发货单，并记录出错信息到 Errors.txt。如果对于某一</li></ol></div>									字段	格式和含义	Item number	字符串型，货物编号	Quantity	整型，货物数量	Supplier	字符串型，供应商编号	Description	字符串型，货物描述
字段	格式和含义																	
Item number	字符串型，货物编号																	
Quantity	整型，货物数量																	
Supplier	字符串型，供应商编号																	
Description	字符串型，货物描述																	

<p>种货物有多个发货单，而且库存总量小于这些发货单的总和的话，系统应该按照发货单中的数量从小到大的有限原则满足客户。也就是说，对于某一种货物如果一个数量 Quantity 少的发货单没有处理之前，数量 Quantity 多的发货单永远不会被处理。（这种处理原则不受发货单记录在 Transactions.txt 的先后顺序影响）</p> <p>6. 以'R'开头的事务表示这是一个到货单记录，在'R'后面是 Item number 和它的数量 Quanlity。处理一条到货单意味着增加库存中相应货物的数量（增加的数量=到货单中的数量）。注意：如果在 Transactions.txt 文件中，到货单出现在发货单之后，到货单中的货物数量可以用来填补发货单中的数量（可以理解成在 Transactions.txt 中，优先处理到货单）。</p> <p>7. 以'A'开头的事务表示向库存中增加一种新的货物（即这种货物以前库存中没有），在'A'后面是 Item number，供应商 supplier 以及货物的描述 description。处理一个新增货物记录意味着向库存中增加一个数量 Quantity 为 0 的新的 Item。你可以假设在一个 Transactions.txt 中，新增货物记录总是出现在第一个到货单之前。</p> <p>8. 以'D'开头的事务表示从库存中删除一种货物，在'D'后面是 Item 号。删除操作总是在所有的事物处理之后才被处理，以保证对于可能出现的同一种货物的发货单的操作能在删除之前被正确处理。如果要删除的某种货物的库存量 Quantity 不为 0 的话，系统应该向 Errors.txt 记录出错信息。</p> <p>9. 文件 Shipping.txt 中的每一行代表给某一客户的发货信息。Shipping.txt 中的每一行分别是客户编号、Item 号、货物数量，它们之间用 tab 键分隔。如果发货单中有两条客户编号和 Item 编号一样的记录，在 Shipping.txt 中应该将这两条发货信息合并（即将它们的数量相加）。</p> <p>10. Errors.txt 文件包含未发送的发货记录和库存量大于 0 的删除记录。Errors.txt 每一行包含 Custom 编号、Item 编号以及发货单上的数量 Quantity。对于删除操作，Custom 编号为 0，数量 Quntity 为库存中的 Quantity。</p> <p>11. 实验测试数据：</p> <div><div>Inventory.txt</div><table><tr><td></td><td>17</td><td>42</td><td>6</td><td>Widget, blue</td></tr><tr><td></td><td>1234</td><td>0</td><td>4</td><td>Whatsit</td></tr><tr><td></td><td>123123</td><td>999999</td><td>98</td><td>Doohicky</td></tr></table></div> <div><div>Transactions.txt</div><table><tr><td>0</td><td>123123</td><td>1000</td><td>9</td><td></td></tr><tr><td>0</td><td>17</td><td>36</td><td>8</td><td></td></tr><tr><td>0</td><td>17</td><td>12</td><td>4</td><td></td></tr><tr><td>R</td><td>123123</td><td>1</td><td></td><td></td></tr><tr><td>D</td><td>1234</td><td></td><td></td><td></td></tr><tr><td>A</td><td>5</td><td>4</td><td></td><td>Thingy</td></tr></table></div> <tr><td><div><div>三、使用硬件、软件环境</div><div>Windows10, 内存 8g, 硬盘 1TB, JDK1.8, eclipse</div><div>四、实验过程、步骤及原始记录(算法、原程序、测试结果，分析等)</div><div>源代码：</div><div>package inven;</div><div>import java.io.*;</div><div>import java.lang.*;</div></div></td></tr>		17	42	6	Widget, blue		1234	0	4	Whatsit		123123	999999	98	Doohicky	0	123123	1000	9		0	17	36	8		0	17	12	4		R	123123	1			D	1234				A	5	4		Thingy	<div><div>三、使用硬件、软件环境</div><div>Windows10, 内存 8g, 硬盘 1TB, JDK1.8, eclipse</div><div>四、实验过程、步骤及原始记录(算法、原程序、测试结果，分析等)</div><div>源代码：</div><div>package inven;</div><div>import java.io.*;</div><div>import java.lang.*;</div></div>
	17	42	6	Widget, blue																																										
	1234	0	4	Whatsit																																										
	123123	999999	98	Doohicky																																										
0	123123	1000	9																																											
0	17	36	8																																											
0	17	12	4																																											
R	123123	1																																												
D	1234																																													
A	5	4		Thingy																																										
<div><div>三、使用硬件、软件环境</div><div>Windows10, 内存 8g, 硬盘 1TB, JDK1.8, eclipse</div><div>四、实验过程、步骤及原始记录(算法、原程序、测试结果，分析等)</div><div>源代码：</div><div>package inven;</div><div>import java.io.*;</div><div>import java.lang.*;</div></div>																																														

```

import java.util.ArrayList;

import org.omg.CORBA.PUBLIC_MEMBER;
class update{//更新库存信息
void upd() throws IOException{

    BufferedReader br = new BufferedReader(
        new FileReader("E:/进货管理/NewInventory.dat"));
    BufferedWriter bw = new BufferedWriter(
        new FileWriter("E:/进货管理/Inventory.dat"));

    String s = null;
    s=br.readLine();
    while(s!=null)
    {
        bw.write(s);
        bw.newLine();
        s=br.readLine();
    }
    bw.close(); //用完要关闭
    br.close(); //用完要关闭
}

}

class global{
    public static int cc=0;
    public static int outcount=0;
}

class updateall{//更新库存信息
    void updall(String file,String path) throws IOException{

        BufferedReader br = new BufferedReader(
            new FileReader(file));
        BufferedWriter bw = new BufferedWriter(
            new FileWriter(path));

        String s = null;
        s=br.readLine();
        while(s!=null)
        {
            bw.write(s);
            bw.newLine();
            s=br.readLine();
        }
        bw.close(); //用完要关闭
        br.close(); //用完要关闭
    }
}

```

```

    }
class updateT{//更新交易记录
    void upd() throws IOException{

        BufferedReader br = new BufferedReader(
            new FileReader("E:/进货管理/NewTransaction.dat"));
        BufferedWriter bw = new BufferedWriter(
            new FileWriter("E:/进货管理/Transaction.dat"));

        String s = null;
        s=br.readLine();
        while(s!=null)
        {
            bw.write(s);
            bw.newLine();
            s=br.readLine();
        }
        bw.close(); //用完要关闭
        br.close(); //用完要关闭
    }
}

class getshortstring{
    String getstring(int i,int j,String s){
        int x=0,y=0,z=0,w=0;
        String shortstr="";
        while(z<i){
            if(s.charAt(x)!='\t'){
                x++;}
            else{
                x++;
                z++;}}
        while(w<j){
            if(s.charAt(y)!='\t'){
                y++;}
            else{
                y++;
                w++;}}
        shortstr=s.substring(x,y-1);
        return shortstr;

    }
}

```

```

class Rscan{//交易记录扫描
void updR() throws IOException{

BufferedReader br = new BufferedReader(
                        new FileReader("E:/进货管理/Transaction.dat"));
BufferedReader brin = new BufferedReader(
                        new FileReader("E:/进货管理/Inventory.dat"));

String s = null;
String invs = null;//对库存行扫描时的变量
String shortstring="";
String shortstringnum="";//记录操作订单的变量
String findinviden="";
String findinvidennum="";
int addnumber=0;
int localnumber=0;
s=br.readLine();
while(s!=null){
    if((s.charAt(0)=='R')){
        getshortstring get=new getshortstring();
        shortstring=get.getstring(1,2,s);//获取进出货项目编号
        shortstringnum=get.getstring(2,3,s);
        addnumber=Integer.parseInt(shortstringnum);
        invs=brin.readLine();
        while(invs!=null){
            findinviden=get.getstring(0,1,invs);
            if(shortstring.equals(findinviden)){
                findinvidennum=get.getstring(1,2,invs);
                localnumber=Integer.parseInt(findinvidennum);
                localnumber=localnumber+addnumber;
                findinvidennum=localnumber+"";

                invs=get.getstring(0,1,invs)+"\t"+findinvidennum+"\t"+get.getstring(2,3,invs)+"\t"+get.getstring(3,4,invs)+"\t";
                //按行修改指定出的代码
                change ch=new change();
                String fileway="E:/进货管理/Inventory.dat";
                String destfileway="E:/进货管理/NewInventory.dat";
                ch.cha(fileway,destfileway,0,1,shortstring , invs);//处理完到货条款
                invs="//"+" \t"+s+"\t";
                fileway="E:/进货管理/Transaction.dat";
                destfileway="E:/进货管理/NewTransaction.dat";
                shortstring="R";
                ch.cha(fileway,destfileway,0,1,shortstring , invs);//删除掉已处理的到货条款
            }
        }
    }
}
}

```

```

        ///

        }
        invs=brin.readLine();
    }
    }

    s=br.readLine();
    }
    br.close(); //用完要关闭
    }
}

class change{
    void cha(String orfile,String destfile,int x,int y,String find,String exchange) throws
IOException{//修改文件地址及特征值
        BufferedReader br = new BufferedReader(
            new FileReader(orfile));
        BufferedWriter bw = new BufferedWriter(
            new FileWriter(destfile));

        String s = null;
        s=br.readLine();
        while(s!=null)
        {   String item;
            getshortstring get=new getshortstring();
            item=get.getstring(x,y,s);
            if(item.equals(find)){
                bw.write(exchange);
                bw.newLine();
                s=br.readLine();
                ///////////////////
            }
            else {
                bw.write(s);
                bw.newLine();
                s=br.readLine();
            }
        }
        bw.close(); //用完要关闭
        br.close(); //用完要关闭
    }
}

```

```

class Ascan{//新增货物记录扫描
    void upd() throws IOException{

        BufferedReader br = new BufferedReader(
            new FileReader("E:/进货管理/Transaction.dat"));
        BufferedReader brin = new BufferedReader(
            new FileReader("E:/进货管理/Inventory.dat"));
        BufferedWriter bwin = new BufferedWriter(
            new FileWriter("E:/进货管理/NewInventory.dat"));

        String s = null;
        String invs = null;//对库存行扫描时的变量
        String shortstring="";
        String shortstringnum="";//记录操作订单的变量
        String findinviden="";
        String findinvidennum="";
        int addnumber=0;
        int localnumber=0;
        s=br.readLine();
        boolean charge=true;
        while(s!=null){
            if((s.charAt(0)=='A')){
                getshortstring get=new getshortstring();
                shortstring=get.getstring(1,2,s);//获取新增货项目编号
                addnumber=Integer.parseInt(shortstring);//新增货物编号转换为 int 型
                invs=brin.readLine();
                charge=true;
                while(invs!=null){
                    findinviden=get.getstring(0,1,invs);
                    localnumber=Integer.parseInt(findinviden);
                    if( (addnumber<localnumber) &&(charge==true)){

                        shortstringnum=get.getstring(1,2,s)+"\t"+"0"+" \t"+get.getstring(2,3,s)+"\t"+get.getstring(3,
                        4,s)+"\t";

                        //////////////////////////////////////

                        bwin.write(shortstringnum);
                        bwin.newLine();
                        bwin.write(invs);
                        bwin.newLine();
                        invs=brin.readLine();
                        charge=false;

                        //////////////////////////////////////
                    }
                }
            }
            s=br.readLine();
        }
    }
}

```

款

```
String ainvs="//"+s+"\t";
String fileway="E:/进货管理/Transaction.dat";
String destfileway="E:/进货管理/NewTransaction.dat";
shortstring="A";
change ch=new change();
ch.cha(fileway,destfileway,0,1,shortstring , ainvs);//删除掉已处理的到货条

updateT tran=new updateT();
tran.upd();//更新交易记录
}
else{
    bwin.write(invs);
    bwin.newLine();
    invs=brin.readLine();
}
}
}
else{};
s=br.readLine();
}
br.close(); //用完要关闭
brin.close();
bwin.close();
}
}

class oscan{
    void updO() throws IOException{
        BufferedReader tran = new BufferedReader(
            new FileReader("E:/进货管理/Transaction.dat"));
        BufferedReader inve = new BufferedReader(
            new FileReader("E:/进货管理/Inventory.dat"));
        //BufferedWriter error = new BufferedWriter(
        //    new FileWriter("E:/进货管理/error.dat"));
        BufferedWriter ninbw = new BufferedWriter(
            new FileWriter("E:/进货管理/NewInventory.dat"));
        //BufferedWriter ship = new BufferedWriter(
        //    new FileWriter("E:/进货管理/Ship.dat"));
        String trans=null,inves=null,gets=null;
        while(global.cc<=global.outcount){
            // outtrans=outtran1.readLine();
            trans=tran.readLine();
            global.cc=global.cc+1;
        }
        //trans=tran.readLine();
```



货条款	<pre>getshortstring get=new getshortstring(); gets=get.getstring(1,2,trans);//获取发出货项目编号 ArrayList itemname=new ArrayList(); ArrayList number=new ArrayList(); ArrayList factory=new ArrayList(); String replace=null; String oriagn=null;  while(trans!=null&amp;&amp;(trans.charAt(0)=='O')&amp;&amp;(itemname.isEmpty()    itemname.indexOf(gets )!=-1))      {   replace=gets;         oriagn=trans;         itemname.add(get.getstring(1,2,trans));         if(factory.indexOf(get.getstring(3,4,trans))!=-1 ){             int j=0,i=0,k=0;             j=factory.indexOf(get.getstring(3,4,trans));             i=(Integer) number.get(j);             k=Integer.parseInt(get.getstring(2,3,trans));             i=i+k;             number.add(i);         }         else{             number.add(get.getstring(2,3,trans));             factory.add(get.getstring(3,4,trans));         }         trans=tran.readLine();         gets=get.getstring(1,2,trans);     }  ////////////////////////////////////     tran.close(); String ainvs="//"+t"+oriagn; String fileway="E:/进货管理/Transaction.dat"; String  destfileway="E:/进货管理/NewTransaction.dat"; String shortstring="O"; ochange cho=new ochange(); cho.ocha(fileway,destfileway,0,1,shortstring , replace,ainvs);//删除掉已处理的出  updateT t=new updateT(); t.upd();//更新交易记录 //////////////////////////////////// inves=inve.readLine(); int min=0,n=1,k=0; String chanship=null; while(inves!=null){</pre>
-----	--

	<pre> //判断库存对应并进行出货处理,,,记得删除已操作的出货记录 int totalnumber=Integer.parseInt( get.getstring(1,2,inves)) ; if(itemname.indexOf(get.getstring(0,1,inves))!=-1){     if(totalnumber&gt;=0){         while(number.size()&gt;0){             min=Integer.parseInt (number.get(0).toString());             while(n&lt;number.size()){                 if(min&gt;Integer.parseInt ( number.get(n).toString())){                     min=Integer.parseInt ( number.get(n).toString());                     n++;                 }                 else{n++;}             }             k=number.indexOf(Integer.toString(min));//获取要减去 的位置              totalnumber=totalnumber-min;             if(totalnumber&gt;0){                 inves= get.getstring(0,1,inves)+"\t"+totalnumber+"\t"+ get.getstring(2,3,inves)+"\t"+ get.getstring(3,4,inves)+"\t";  chanship=factory.get(k).toString()+"\t"+itemname.get(0).toString()+"\t"+number.get(k).toString() +"\t";                  BufferedReader ship1= new BufferedReader(                     new FileReader("E:/进货管理 /ship.dat"));                  BufferedWriter newship = new BufferedWriter(                     new FileWriter("E:/进货管理 /newship.dat"));                  String ships = null;                 ships=ship1.readLine();                 while(ships!=null)                 {                     newship.write(ships);                     newship.newLine();                     ships=ship1.readLine();                 }                 newship.write(chanship);                 newship.newLine();                 ship1.close(); //用完要关闭                 newship.close(); //用完要关闭                 updateall allship=new updateall();                 allship.updall("E:/进货管理/newship.dat", "E:/进货管理 </pre>
--	--

```

/ship.dat");

        }
        else{
            totalnumber=totalnumber+min;

            chanship=factory.get(k).toString()+"\t"+itemname.get(0).toString()+"\t"+number.get(k).toString()+"\t";

            BufferedReader error1= new BufferedReader(
                new FileReader("E:/进货管理
/error.dat"));

            BufferedWriter newerror = new BufferedWriter(
                new FileWriter("E:/进货管理
/newerror.dat"));

            String errors = null;
            errors=error1.readLine();
            while(errors!=null)
            {
                newerror.write(errors);
                newerror.newLine();
                errors=error1.readLine();
            }
            newerror.write(chanship);
            newerror.newLine();
            error1.close(); //用完要关闭
            newerror.close(); //用完要关闭
            updateall allerror=new updateall();
            allerror.updall("E:/进货管理/newerror.dat", "E:/进货管理
/error.dat");

            }
            number.remove(k);
            factory.remove(k);
            itemname.remove(k);

        }
        // ninbw.write(inves);
        // ninbw.newLine();
    }
}
if(inves!=null){
    ninbw.write(inves);
    ninbw.newLine();
    inves=inve.readLine();
}
}
}//////,,

```

```

        ninbw.close();
        //error.close();
        //ship.close();
        update inv =new update();
        inv.upd();//更新库存记录
    }
}
class ochange{
    void ocha(String orfile,String destfile,int x,int y,String find,String find2,String exchange)
throws IOException{//修改文件地址及特征值
        BufferedReader br = new BufferedReader(
            new FileReader(orfile));
        BufferedWriter bw = new BufferedWriter(
            new FileWriter(destfile));
        String s = null;
        s=br.readLine();
        while(s!=null)
        {   String item;
            getshortstring get=new getshortstring();
            item=get.getstring(x,y,s);
            String item2=get.getstring(1,2,s);
            if(item2.equals(find2)){
            if(item.equals(find)){
                bw.write(exchange);
                bw.newLine();
                s=br.readLine();
                //////////////////////////////////
            }
            else {
                bw.write(s);
                bw.newLine();
                s=br.readLine();
            }
        }
        else{ bw.write(s);
            bw.newLine();
            s=br.readLine();};
        }
        bw.close(); //用完要关闭
        br.close(); //用完要关闭
    }
}
class delte{
    void del() throws IOException{

```

```

BufferedReader delbtr = new BufferedReader(
    new FileReader("E:/进货管理/Transaction.dat"));
String delbtrs=null,dshort=null,invds=null;
delbtrs=delbtr.readLine();
String rep=null;
while(delbtrs!=null)
{
    if(delbtrs.charAt(0)=='D'){
        rep=delbtrs;
        getshortstring getd=new getshortstring();
        dshort=getd.getstring(1,2,delbtrs);
        BufferedReader invd = new BufferedReader(
            new FileReader("E:/进货管理/Inventory.dat"));
        BufferedWriter inb = new BufferedWriter(
            new FileWriter("E:/进货管理/NewInventory.dat"));
        invds=invd.readLine();
        while(invds!=null){
            if(dshort.equals( getd.getstring(0,1,invds))) {
                if(getd.getstring(1,2,invds).equals("0")){
                    invds=invd.readLine();
                }
            }
            else{

                inb.write(invds);
                inb.newLine();
                ///////////////出错处理
                BufferedReader error1= new BufferedReader(
                    new FileReader("E:/进货管理/error.dat"));
                BufferedWriter newerror = new BufferedWriter(
                    new FileWriter("E:/进货管理/newerror.dat"));

                String errors = null;
                errors=error1.readLine();
                while(errors!=null)
                {
                    newerror.write(errors);
                    newerror.newLine();
                    errors=error1.readLine();
                }

                newerror.write("0"+"\\t"+getd.getstring(0,1,invds)+"\\t"+getd.getstring(1,2,invds)+"\\t");
                newerror.newLine();
                error1.close(); //用完要关闭
                newerror.close(); //用完要关闭
                updateall allerror=new updateall();
            }
        }
    }
}

```

款

```
allerror.updall("E:/进货管理/newerror.dat", "E:/进货管理/error.dat");
////////////////////////////////////
invds=invd.readLine();
    }
}
else{inb.write(invds);
inb.newLine();
invds=invd.readLine();
}
}
invd.close();
inb.close();
}

delbtrs=delbtr.readLine();
}
delbtr.close();
String ainvs="//"+rep+"\t";
    String fileway="E:/进货管理/Transaction.dat";
    String destfileway="E:/进货管理/NewTransaction.dat";
    String shortstring="D";
    change ch=new change();
    ch.cha(fileway,destfileway,0,1,shortstring , ainvs);//删除掉已处理的到货条

    updateT tran=new updateT();
    tran.upd();//更新交易记录
}
}

public class Inventory {
    public static void main(String[] args)throws Exception{
        Rscan find=new Rscan();
        find.updR();//完成货物补充
        updateT tran=new updateT();
        tran.upd();//更新交易记录
        update inv =new update();
        inv.upd();//更新库存记录
        Ascan add=new Ascan();
        add.upd();
        tran.upd();//更新交易记录
        inv.upd();//更新库存记录
        // int cc=0;
        BufferedReader outtran = new BufferedReader(
            new FileReader("E:/进货管理/Transaction.dat"));
        String outtrans=null;
```

```

        outtrans=outtran.readLine();
//    int outcount=0;
        outtran.close();
        getshortstring outget=new getshortstring();
        String outshort=outget.getstring(0,1,outtrans);
        while(outtrans!=null){
            if(outshort.equals("O")){
                oscan orpe=new oscan();
                orpe.updO();
                global.outcount=global.outcount+1;
            }
            else{global.outcount=global.outcount+1;}
        }
        BufferedReader outtran1 = new BufferedReader(
            new FileReader("E:/进货管理/Transaction.dat"));
        while(global.cc<=global.outcount+1){
            outtrans=outtran1.readLine();
            global.cc=global.cc+1;
        }
        global.cc=0;
        outtran1.close();
        if(outtrans!=null){
            outshort=outget.getstring(0,1,outtrans);
        }
    }

    delte de=new delte();
    de.del();
    update inve =new update();
    inve.upd();//更新库存记录
    System.out.println("处理完毕， 请查看结果");
}
}

```

**实验前：**

Inventory.dat - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
17      42      6      Widget, blue
1234    1       4      whatsit
123123  999999  98     doohicky
```

Transaction.dat - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
0      123123  1000   9
O      17     36     8
O      17     12     4
R      123123  1
D      1234
A      5       4      Thingy
```

运行后:



Inventory.dat - 记事本

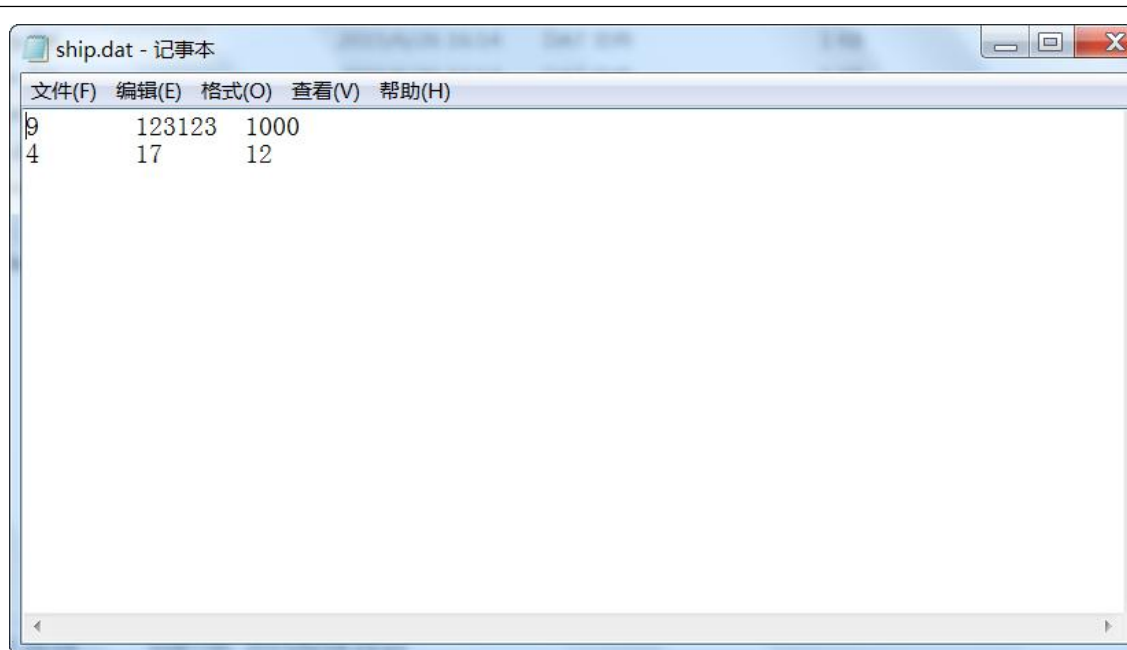
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

5	0	4	Thingy
17	30	6	Widget, blue
1234	1	4	whatsit
123123	999000	98	doohicky

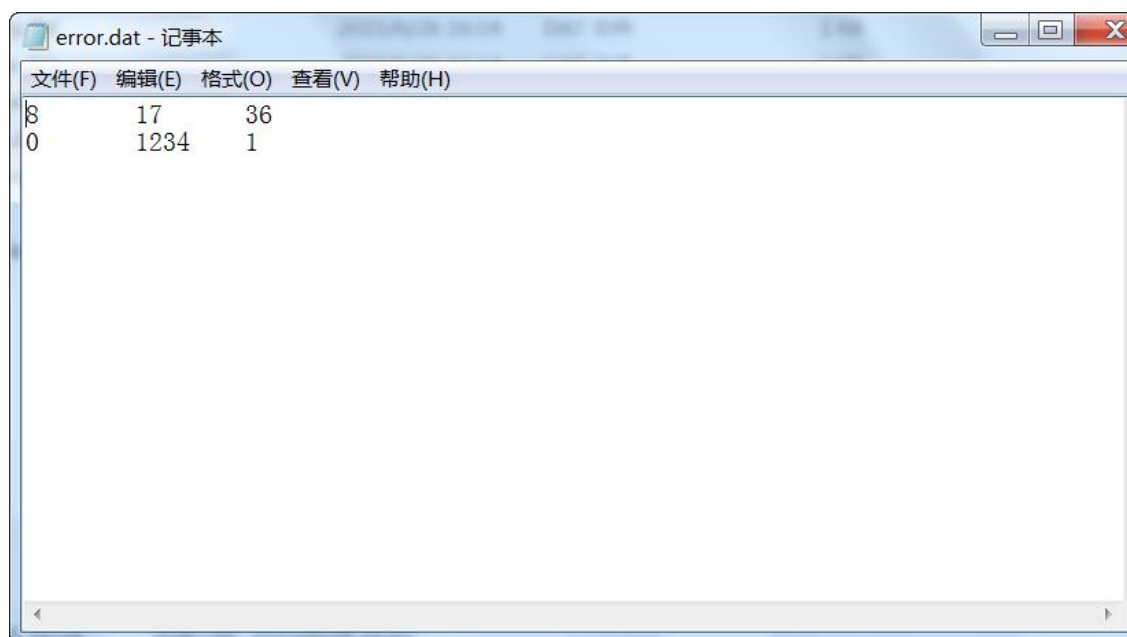
Transaction.dat - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

//	O	123123	1000	9
//	O	17	12	4
//	O	17	12	4
//	R	123123	1	
//	D	1234		
//	A	5	4	Thingy



```
ship.dat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
9      123123  1000
4      17      12
```



```
error.dat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
8      17      36
0      1234    1
```

## 五、实验结论、分析、思考题与心得体会

1. 掌握了 Java 中文件的 `BufferedReader` 和 `BufferedWriter` 实现输入输出信息的读写操作。
2. 学会了使用 Java 提供的实用类 (`Vector`, `ArrayList`) 来完成特定的功能。
3. 掌握了字符串类 (`String`, `StringBuffer`) 的使用。
4. 掌握了用面向对象的方法分析和解决复杂问题。

