package CourseProject**;**import java.io.BufferedWriter**;**import java.io.File**;**import java.io.FileWriter**;**import java.io.IOException**;**import java.text.ParseException**;**import java.text.SimpleDateFormat**;**import java.util.\***;**public class DeviceManager {  
 public static void main(String[] args) throws IOException**,** ParseException {  
  
  
 //创建  
 ArrayList<Device> arrayList=new ArrayList<>()**;** while (true){  
 System.*out*.println("-----欢迎来到主界面-----")**;** System.*out*.println("1.录入设备消息")**;** System.*out*.println("2.修改设备消息")**;** System.*out*.println("3.查询设备消息")**;** System.*out*.println("4.分类统计设备")**;** //System.out.println("5.查询设备消息");  
 System.*out*.println("6.退出")**;** System.*out*.print("请输入你的选择：")**;** Scanner sc=new Scanner(System.*in*)**;** String line=sc.nextLine()**;** switch (line){  
 case "1":  
 *addDevice*(arrayList)**;** break**;** case "2":  
 *readdDevice*(arrayList)**;** break**;** case "3":  
 *allDevice*(arrayList)**;** break**;** case "4":  
 *numberDevice*(arrayList)**;** break**;** case "6":  
 System.*out*.println("谢谢使用")**;** System.*exit*(**0**)**;**//jvm退出  
  
 }  
 }  
 }  
 private static File *file*=new File("d:\\DeviceProject.txt")**;** private static SimpleDateFormat *sdf*=new SimpleDateFormat("yyyy-MM-dd")**;** public static void addDevice(ArrayList<Device> array) throws IOException{  
 //可以while 多次输入  
 Scanner sc=new Scanner(System.*in*)**;** System.*out*.println("请输入设备编号")**;** String id=sc.nextLine()**;** System.*out*.println("请输入设备种类")**;** String series=sc.nextLine()**;** System.*out*.println("请输入设备名称")**;** String name=sc.nextLine()**;** System.*out*.println("请输入设备价格")**;** String price=sc.nextLine()**;** System.*out*.println("请输入设备购入时间,例:yyyy-MM-dd")**;** *sdf*.setLenient(false)**;**//提示格式错误  
 Date intime= null**;** try {  
 intime = *sdf*.parse(sc.next())**;** } catch (ParseException e) {  
 System.*out*.println("日期格式错误")**;** return**;** }  
 System.*out*.println("请输入设备报废时间")**;** Date outtime= null**;** try {  
 outtime = *sdf*.parse(sc.next())**;** } catch (ParseException e) {  
 System.*out*.println("日期格式错误")**;** return**;** }  
  
  
 Device de=new Device(id**,** series**,** name**,**price**,** intime**,** outtime)**;** if(!*file*.exists()){  
 *file*.createNewFile()**;** }  
 //考虑编号相同  
 for(Device device:array){  
 if(de.getId().equals(id)){  
 System.*out*.println("编码重复,重新输入")**;** return**;** }  
 }  
  
 array.add(de)**;** try (FileWriter fw = new FileWriter(*file***,** true)**;** // 使用true参数追加内容到文件  
 BufferedWriter bw = new BufferedWriter(fw)) {  
 for (Device device : array) {  
 bw.write(device.toString())**;** bw.newLine()**;** // 换行  
 }  
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
  
// FileWriter fw=new FileWriter(file,true);  
// BufferedWriter bw=new BufferedWriter(fw);{  
// for(Device device:array){  
// bw.write(de.toString());  
// bw.newLine();//换行符  
// }  
// }  
  
// bw.write(String.valueOf(de));  
// bw.newLine();  
  
  
 // bw.flush();  
 //bw.close();  
  
 }  
  
  
 public static void readdDevice(ArrayList<Device> array) throws ParseException**,** IOException {  
 Scanner in=new Scanner(System.*in*)**;** System.*out*.print("输入需要修改设备编码：")**;** String id**;** id= in.nextLine()**;** System.*out*.print("输入新的编号：")**;** String sid=in.nextLine()**;** System.*out*.print("输入新的种类：")**;** String series=in.nextLine()**;** System.*out*.print("输入新的名称：")**;** String name=in.nextLine()**;** System.*out*.print("输入新的价格：")**;** String price=in.nextLine()**;** SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd")**;** System.*out*.println("请输入设备购入时间,例:yyyy-MM-dd")**;**// Calendar start=Calendar.getInstance();  
// start.set(sc.nextInt(),sc.nextInt(),sc.nextInt());  
 Date intime=sdf.parse(in.next())**;** System.*out*.println("请输入设备报废时间")**;** Date outtime=sdf.parse(in.next())**;** // String date2=sc.nextLine();  
 //Date date2=sdf.parse(sc.nextLine());  
 // DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");  
 // LocalDateTime dateTime=sdf.parse(outtime);  
  
 Device de=new Device(id**,** series**,** name**,**price**,** intime**,** outtime)**;**//否则用set  
 array.add(de)**;** FileWriter fw=new FileWriter(*file***,**true)**;** BufferedWriter bw=new BufferedWriter(fw)**;**// for(Device device:array){  
// bw.write(de.toString());  
// bw.newLine();//换行符  
// }  
 for(int i=**0;**i<array.size()**;**i++){  
 Device device = array.get(i)**;** if(device.getId().equals(id)){  
 bw.write(de.toString())**;** bw.newLine()**;**//换行符  
 //array.set(i,de);//索引存入  
 break**;** }  
 }  
 bw.flush()**;** bw.close()**;** }  
 public static void allDevice(ArrayList<Device> array){  
 //判断是否有数据  
 if(array.size()==**0**){  
 System.*out*.println("请先录入信息")**;** return**;**//为了让程序不往下执行  
 }  
 System.*out*.println("设备编号 \t\t种类 \t名称 \t价格 \t购入时间 \t\t报废时间 \t\t ")**;** for(Device i:array)  
 {  
 //Device device=array.get(i);  
 System.*out*.println(i)**;** }  
}  
 public static void numberDevice(ArrayList<Device> array){  
 Date date=new Date()**;** TreeMap<String**,** Boolean > 报废设备=*delectDevice*(date**,**array)**;** for(Map.Entry<String**,**Boolean> entry : 报废设备.entrySet()){  
 if(entry.getValue()){  
  
 System.*out*.println("报废设备名称："+entry.getKey())**;** }  
 }  
 //统计每个设备数量  
 Map<String**,**Integer> deviceCount=new HashMap<>()**;** for(Device device:array){  
 String deviceName = device.getName()**;** deviceCount.put(deviceName**,**deviceCount.getOrDefault(deviceName**,0**)+**1**)**;** }  
 //输出2啊  
 for(Map.Entry<String**,**Integer> entry:deviceCount.entrySet())//  
 {  
 System.*out*.println("设备名称："+entry.getKey()+",设备数量："+entry.getValue())**;** }  
  
 }  
 public static TreeMap<String**,** Boolean > delectDevice(Date date**,** ArrayList<Device> array){  
  
 TreeMap<String**,** Boolean > 报废设备=new TreeMap<>()**;** for(Device device:array){  
 if(device.getOuttime().after(date)){  
 报废设备.put(device.getName()**,**true)**;** }  
 else {  
 //如果设备存在,不覆盖  
 报废设备.putIfAbsent(device.getName()**,** false)**;** }  
 }  
 return 报废设备**;**// for(int i=0;i< array.size();i++){  
//  
// if(device.getOuttime().after(date)){  
// f=false;  
//  
// }  
// }  
 }  
  
  
}

package CourseProject**;**import javax.xml.crypto.Data**;**import java.text.SimpleDateFormat**;**import java.util.Date**;**/\*  
 主界面  
AIT+Insert  
 \*/  
public class Device {  
 private String id**;** private String series**;** private String name**;** private Date intime**;** private Date outtime**;**private String price**;** public Device() {  
 }  
  
 public Device(String sid**,** String series**,** String name**,** String price**,** Date intime**,** Date outtime) {  
 this.id = sid**;** this.name = name**;** this.intime = intime**;** this.outtime = outtime**;** this.price=price**;** this.series=series**;** }  
  
 @Override  
// public String toString() {  
// SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");  
// return "设备编号："+id+"\t\t种类:"+series+"\t名称:"+name+"\t\t价格:"+price+"\t购入时间:"+sdf.format(intime)+"\t\t报废时间:"+sdf.format(outtime);//不存入怎么行  
// }  
 public String toString() {  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd")**;** return "Device{" +  
 "id='" + id + '\'' +  
 ", series='" + series + '\'' +  
 ", name='" + name + '\'' +  
 ", price='" + price + '\'' +  
 ", intime='" + sdf.format(intime) + '\'' +  
 ", outtime='" + sdf.format(outtime) + '\'' +  
 '}'**;** }  
  
 public Date getIntime() {  
 return intime**;** }  
  
 public void setIntime(Date intime) {  
 this.intime = intime**;** }  
  
 public Date getOuttime() {  
 return outtime**;** }  
  
 public void setOuttime(Date outtime) {  
 this.outtime = outtime**;** }  
  
 public String getSeries() {  
 return series**;** }  
  
 public void setSeries(String series) {  
 this.series = series**;** }  
  
 public String getId() {  
 return id**;** }  
  
 public void setId(String id) {  
 this.id = id**;** }  
  
 public String getName() {  
 return name**;** }  
  
 public void setName(String name) {  
 this.name = name**;** }  
  
  
  
 public String getPrice() {  
 return price**;** }  
  
 public void setPrice(String price) {  
 this.price = price**;** }  
}