

Task1

Q: 设计一个程序 实现功能: 将doro.jpg复制一份到项目根目录下 新图片命名为doro_copy.jpg。

A: 程序参见task9/src/Task1.java。其中File对象表示对一个文件/目录的引用，这体现了Java面向对象的特点。

读取和写入文件使用了字节流，并且使用数组作为缓冲，一次性读入多个字节，能避免一次性读取大文件导致内存占用飙升的情况。

FileInputStream类的read方法在返回-1时表示文件以及读取完成，这是一种利用返回值来判断程序运行状态的思想，利用这样的返回值我们可以随时知道程序读取文件的状态。

至于字节流需要关闭的问题，这一点很多语言都差不多，使用完了必须关闭它们来释放资源。不过java的try-with-resources语法糖可以自动关闭实现了closeable接口的对象（这里使用的两个FileStream也不例外）。

Task2

Q: 设计一个程序: 读取name.txt 将表中的人名按照unicode自然排序 每个人名占据一行 在项目根目录下创建name_sorted.txt文件 将排序后的内容写入name_sorted.txt文件 要求去掉多余的首尾空格与空行。

A: 程序参见task9/src/Task2.java。其中读取和写入分别使用了3个流，一个用于输入/出，一个用于转换编码，一个用于缓冲。它们各司其职，完成了这个工作。

关闭流的时候可以像task9的Task1一样用try-with-resources来关闭。

添加换行符的时候，使用newLine()方法而不是用换行符是为了保持程序在各个环境下能够正常运行。不同系统的换行符不太相同:

系统	换行符
Windows	\r\n
Unix/Linux	\n
MacOS	\r

而用newLine()方法，程序员就不需要关心对应的操作系统也能正确换行了，这也体现了Java“一次编译，到处运行”的口号。

Task3

Q: 用下面的代码 创建一个student对象 使用序列化将该对象的信息储存在项目文件根目录下的student.dat文件中 再使用反序列化将数据赋值给新的Student对象doro 最后输出doro的信息 (Student的toString方法已重写好了)

```
public class Student {
    private Integer id;
    private String name;
    private Integer gender; //1表示男性 2表示女性
    private String phone;

    public Student(Integer id, String name, Integer gender, String phone) {
        this.id = id;
        this.name = name;
        this.gender = gender;
        this.phone = phone;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Integer getGender() {
        return gender;
    }

    public void setGender(Integer gender) {
        this.gender = gender;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    @Override
```

```
public String toString() {  
    return "Student{" +  
        "id=" + id +  
        ", name='" + name + '\'' +  
        ", gender=" + gender +  
        ", phone='" + phone + '\'' +  
        '}';  
}  
}
```

A: 程序参见task9/src/Task3.java和task9/src/Student.java。

实现对象的序列化主要是看对应的类是否“实现”了Serializable接口。这个接口里面没有定义任何方法，单纯只是为了做标记，声明这个类和相应的对象可以被序列化。

序列化和反序列化主要靠ObjectOutputStream和ObjectInputStream并传入相应的输入/出流以及相应的对象。

值得注意的是，序列化保存出来的文件是二进制文件而非文本文件。