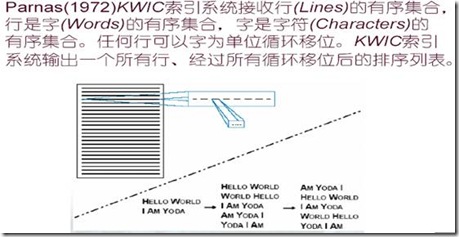
1.问题描述

设计并实现一个软件系统。要求如下：

主要功能如下图所示：



该软件系统由命令行指定要进行处理的英文文本文件（.txt）的目录和文件名以及输出结果文件的目录和文件名。

2.解决方案：

基本面向对象方案：

该方案是个基本的面向对象方案，有四个基本功能类：input, shift, alphabetize 和 output，将这些功能在主程序中串联起来，通过共享数据的方式组成整个系统。

流程很简单，首先通过input类设定文件输入输出路径并且读入要处理的文件，接着使用shift进行移位操作，得到一行行的结果，此时再采用alphabetizer对这些行进行首字母排序，最后采用output类输出。

评价：将各个功能模块分开设计，并且利用权限控制保证了封装的特性，系统的扩展性和封装性增强。

基于事件通知的方案：

基于事件通知的系统中的各个功能并不是直接被调用的，而是通过组建通知或广播事件信息而触发的。其他组件可以通过注册一个与某个事件通知相关联的过程而与其发生联系。例如，我们的IDE就是一个典型的这样的系统，编辑器和变量监视器与调试器的断点功能相联系，当调试器停留在某一个断点的时候，它就会自动去通知与其注册的组件，去定位代码或者观察变量的取值。

一般这样的方案有两种设计，一个是系统有一个分立出来的事件中心，专门负责接收所有传来的信息，并且将它们分发给系统的其他组件，它可以是广播，可以是针对某些特定事件而设定特定的反应。

一种方案不设置一个集权的事件中心，每个模块允许其他模块在他们发送的消息中声明他们需要干什么，这样每次模块发送的消息都只发送给对这个事件感兴趣的模块去，而无需发送到事件中心。

两个行存储模块，第一个行存储模块负责存储所有原先的行，第二个行存储模块负责存储所以循环移位后的模块。 输入模块负责设定路径并且从输入文件中读入并且存储到第一个行存储模块中。 循环移位模块负责循环移位并且存储在第二个行存储模块。输出模块负责输出到文件，主控模块负责主控流程。添加一行到第一个行存储模块就触发了一个事件向循环移位模块发送，循环移位模块会根据这个事件对这一行就行处理，存储到第二个行存储模块，这就会另一个事件发送到排序模块，模块会把新加进来的行与旧行进行混合然后排序。

实现的是Observable/Observers模式，其中的事件互动有两部分：

移位循环模块和第一个行存储模块注册。移位模块是第一个行存储模块的观察者。

排序模块注册和第二个行存储模块注册。排序模块是第二个行存储模块的观察者。