Q&A

### Assignment 1:

1. **Which modules from the original KWIC system did you need to modify to implement the new data representation of circular shifts? What conclusion can you draw here? How does the KWIC system implemented by means of main/subroutine architecture with shared data withstand design changes in data representation?**

**A:**

* 需要修改circularShift和alphabetizing两个方法的实现
* 各个模块之间共享了一些数据结构导致耦合过紧，修改某个数据结构会影响到其他模块，牵一发而动全身。
* KWIC系统的当前实现无法应对设计变化的原因就是各个模块之间共享一些数据结构，导致模块之间高耦合。通过将共享数据结构的方式改为参数传递的方式，使得对于数据结构的修改只在模块内部产生影响，对外的函数声明仍然保持不变或者只在必要时改变。

1. **Which modules from the original KWIC system did you need to modify to add filtering function to the system? Was it difficult to add a new processing component (filter function) accessing the shared data? Can we conclude here that the KWIC system with main/subroutine architecture with shared data is robust to design changes in system's function (assuming that new functions access the shared data)?**

**A:**

* 在当前KWIC系统中添加一个filter模块不需要修改其他的部分。
* 若要在当前KWIC系统中添加新的模块，由于各个模块之间有许多共享数据结构，在添加时需要考虑新模块所修改的数据结构是否会影响其他模块的实现，若有影响，则需要修改相关模块的实现，否则无需修改其他模块
* 基于共享数据结构的main/subroutine架构面对设计的变化并不健壮，修改任何一个涉及共享数据结构的模块都需要考虑所做的修改对其他模块的影响，在系统的功能模块越来越庞杂的情况下，修改或添加/删除一个模块是非常痛苦的。

1. **Would it be difficult to reuse modules from the existing KWIC system in other systems? For example, suppose that in another software system we need to sort a number of lines. Suppose also that in this system lines are represented differently than in the existing KWIC system. Could we reuse the existing alphabetizing function as it is, or do we need to make some modification to the exisiting code to be able to reuse it?**

**A:**

在其他系统中复用当前系统中的模块有两种常用方法：

* + 直接修改现有的模块，修改其中涉及的数据结构和实现以适应新的系统，其工作量接近于重新实现一个相同功能的模块
  + 应用“适配器模式”（Adapter Pattern），在现有模块基础上封装一层类，使该类提供的调用接口适应新的系统，虽然可以屏蔽模块内部的变化，但是仍需要额外的工作量

由此看出，在模块间使用共享数据结构的系统中，任何的修改都可能产生新的工作量，模块间耦合过紧，系统复杂难以维护，模块难以复用。

### Assignment 2:

1. **Which modules from the original KWIC system did you need to modify to implement the new data representation of lines within the LineStorage class? What conclusion can you draw here? How does the KWIC system implemented by means of object-oriented architecture withstand design changes in data representation within a particular module? Is this true for any other object-oriented system?**

**A:**

* 修改LineStorage类中表示line的数据结构不涉及对原始系统中其他模块的修改
* 面向对象的编程方法将类的实现进行了封装，使接口和实现分离，也就屏蔽了类的实现的变化，只要类对外的接口不变，修改类的实现就不影响其他的模块；若必须修改类的对外接口，只要符合“面向接口编程”的原则，也只需要做少量的修改。
* 当前系统通过对不同功能模块封装成类，屏蔽了类之间的变化，数据结构通过参数传递来共享，以此保证了系统的健壮性。但也存在一些问题，例如为了使一个对象和另一个对象通进行交互，必须知道对象的标识，只要一个对象的标识变了，就必须修改所有其他明确调用它的对象（这一问题可以通过使用带有refactor功能的IDE例如Eclipse来完成）；另外调用同一个对象的不同对象之间并不知悉它们中的其他对象将对被调用的共同对象做出什么修改。

1. **Which modules from the original KWIC system did you need to modify to implement a new processing algorithm in the system? Can we conclude here that the KWIC system with object-oriented architecture is robust to design changes in processing algorithm of the system?**

**A:**

* 需要修改KWIC类中的main和execute方法
* 可以认为面向对象的体系结构面对设计的变化时是健壮的

1. **Would it be difficult to reuse modules from the existing KWIC system in other systems? For example, suppose that in another software system we need to sort a number of lines. Suppose that the sorting algorithm expects lines to be represented as arrays of characters. Can you describe a possible object-oriented solution to reuse the LineStorage class without need to modify the source code of the class.**

**A:**

* 在其他系统中复用现存系统中的模块是相对简单的，只需要调用LineStorage.getLineAsString()方法得到line的String表示，然后调用String.toCharArray()方法即可把line的String表示转换成字符数组表示

### Assignment 3:

1. **Which modules from the original KWIC system did you need to modify to implement the WordsIndex module? What can you conclude here? Do systems implemented with event based architecture allow for easy adding of new modules, assuming that the new modules apply the already existing event protocol?**

**A:**

* 只需要添加一个WordsIndex类并修改KWIC类中的main和execute方法
* 可以总结出在“观察者”模式的架构下新模块的添加和修改非常方便
* 添加新的模块只需要实现新的模块类，并添加/修改相应的事件处理方法，几乎不用对现有系统进行修改

1. **Please, describe the easiest way to implement a WordsIndex that should keep all words that appear in the original lines and the number of their occurrences in the circular shifts.**

**A:**

* 创建WordsIndex类继承Observer类
* 重载父类的update方法，在方法中添加对ADD和DELETE事件的响应代码
* 在新line添加的时候出触发ADD事件，统计出当前添加的line中words各自的出现次数
* 在删除一个line的时候出发DELETE事件，并修改该line中words在WordsIndex中的统计次数
* 最后在主控类KWIC的execute方法中令WordsIndex的对象监听LineStorageWraper的事件

1. **One of the main properties of event based systems is so-called "implicit" invocation of procedures. That means some of the procedures in the system are not invoked directly, but rather an event is sent into the system, and then the system itself after processing the events invokes these procedures. Now, taking into account that the processing of events and deciding which procedures to call might be quite complex in systems with a large number of modules and a complex event protocol what can you conclude about the performance of event based system? Do you think that these systems may achieve the same level of the performance as systems with direct procedure calls?**

**A:**

* 隐式调用的事件机制主要优点是为软件重用提供了强大的支持，为改进系统带来了方便，当一个模块被另一个模块替换时不会影响到其他模块。其缺点是，模块放弃了对系统的控制，一个事件发生的时候不确定有哪些模块会响应它，也不能保证这些事件响应方法被调用的顺序；有些情况下基于事件的系统必须靠一个共享数据结构进行交互，这时全局性能和资源管理便成了问题；事件响应过程的语义必须依赖于被触发事件的上下文约束，其关于正确性的推理存疑。
* 在事件并不庞杂、事件之间逻辑较为清晰的系统中，使用事件机制可以大大的提高系统性能，因为在最好的情况下对每个事件的响应可以并行处理，相较于直接调用过程的体系结构编程风格较为清晰，代码修改和重构更方便，模块的复用性也更好。

### Assignment 4:

1. **In both modifications of the current system you needed to extend the functionality of the system. Which modules from the original KWIC system did you need to modify to implement the new functionality in both cases? Does this mean that the pipe and filter KWIC system is robust to design changes in the functionality of the system? Is the same true for any pipe and filter system?**

**A:**

* 只需要修改主控类KWIC类的execute方法将新添加的两个过滤器按照特定顺序添加到处理过程中，修改相应的管道传递顺序即可。
* 该KWIC系统的管道/过滤器架构面对设计的变化并不十分健壮，当系统中过滤器数量增大后，在一次添加多个过滤器的情况下，每次添加都要对插入位置的上下行代码进行修改，以改变参数传递顺序，删除过滤器的时候也会出现相同的情况。
* 在其他的管道/过滤器系统中并不一定存在同样的问题，但也会有一些共性的问题存在，例如该种机制使得进程成为批处理的结构，因为程序员必须将每个过滤器看成一个完整的从输入到输出的转换，在数据传输上也没有通用的标准，每个过滤器都增加了解析和合成数据的工作，导致系统性能下降，也增加了编写过滤器的复杂性。

1. **Filters in pipe and filter systems are completely independent entities. Thus, filters:**
2. **Do not share state (data) with other filters.**
3. **Do not know the identity of neighbor filters in the pipeline.**
4. **Process their data concurrently and incrementally, i.e., they do not wait for other filters to start or finish their jobs.**

**Would it be possible to implement an interactive version of the KWIC system, which will allow users to delete certain lines from the data flow but still not violate any of the above defined filter properties?**

**A:**

* 不太可能，在不共享数据结构的情况下，要想增量地改变所处理的数据，必须要保证过滤器之间的处理顺序，因此不能同步地处理。

1. **Normally, filters in pipe and filter systems work as follows. They read data from the input pipe, transform it and write it to the output pipe. Usually, this involves copying of data from the input to the output pipe. If the number of filters in a pipeline of a complex pipe and filter system is large then there is a lot of in-memory data copying. How this may affect the overall system performance?**

**A:**

* 由于每个过滤器都要独立地对数据进行解析和合成的工作，在过滤链上的过滤器过多的时候这些重复的操作所消耗的资源是可观的，势必影响整个系统的性能。