5.3 封装

耦合和聚合的概念：

要评判某些设计比其他的设计优秀，就得定义一些在类的设计中重要的术语，以用来讨论 设计的优劣。对于类的设计来说，有两个核心术语：耦合和聚合。 耦合这个词指的是类和类之间的联系。之前的章节中提到过，程序设计的目标是一系列通 过定义明确的接口通信来协同工作的类。耦合度反映了这些类联系的紧密度。我们努力要获得 低的耦合度，或者叫作松耦合（loose coupling）。

耦合度决定修改应用程序的容易程度。在一个紧耦合的结构中，对一个类的修改也会导致 对其他一些类的修改。这是要努力避免的，否则，一点小小的改变就可能使整个应用程序发生 改变。另外，要想找到所有需要修改的地方，并一一加以修改，却是一件既困难又费时的事情。 另一方面，在一个松耦合的系统中，常常可以修改一个类，但同时不会修改其他类，而且 整个程序还可以正常运作。

本周会讨论紧耦合和松耦合的例子。 聚合与程序中一个单独的单元所承担的任务的数量和种类相对应有关，它是针对类或方法 这样大小的程序单元而言的理想情况下，一个代码单元应该负责一个聚合的任务（也就是说，一个任务可以被看作是 一个逻辑单元）。一个方法应该实现一个逻辑操作，而一个类应该代表一定类型的实体。聚合 理论背后的要点是重用：如果一个方法或类是只负责一件定义明确的事情，那么就很有可能在 另外不同的上下文环境中使用。遵循这个理论的一个额外的好处是，当程序某部分的代码需要 改变时，在某个代码单元中很可能会找到所有需要改变的相关代码段。

5.4 5.5

可扩展性

从程序中识别出框架和数据，以代码实现框架，将部分功能以数据的方式加载，这样能在很大程度上实现可扩展性。