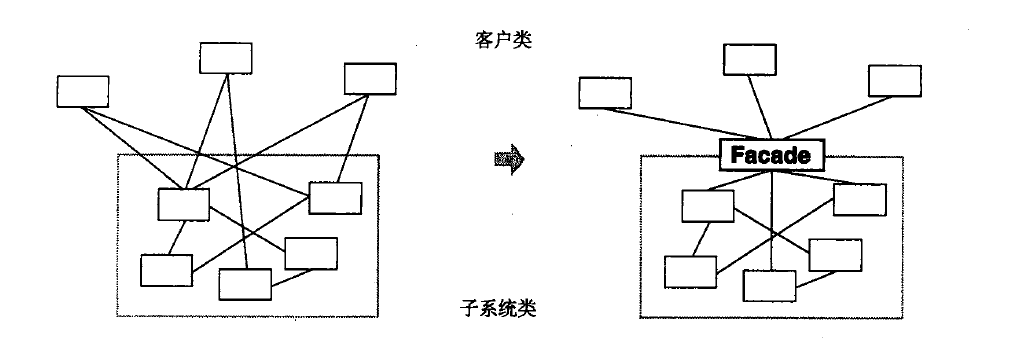
**Facade（外观）**

意图

为子系统中一组接口提供一个一致的界面，Façade模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

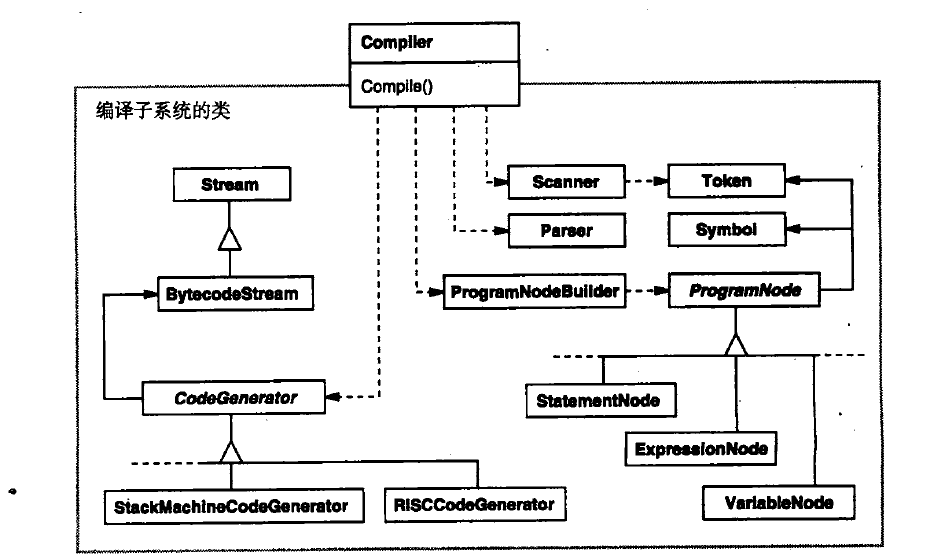
动机

将一个系统划分成为若干个子系统有利于降低系统的复杂性。一个常见的设计目标是使子系统的通信和相互依赖关系达到最小。达到该目标的途径之一是就是引入一个外观对象，它为子系统中较一般的设施提供了一个单一而简单的界面。



例如有一个编程环境，它允许应用程序访问它的编译子系统。这个编译子系统包含了若干个类，如Scanner、Parser、Program Node、BytecodeStream和ProgramNodeBuilder，用于实现这一编辑器。有些特殊应用程序需要直接访问这些类，但是大多数编译器的用户并不关心语法分析和代码生成这样的细节；他们只是希望编译一些代码。对这些用户，编译子系统中那些功能强大但层次较低的接口只会使他们的任务复杂化。

为了提供一个高层的接口并且对客户屏蔽这些类，编译子系统还包括一个Complier类。这个类定义了一个编译器功能的统一接口。Complier类是一个外观，它给用户提供了一个单一而简单的编译子系统接口。它无需完全隐藏实现编译功能的那些类，即可将它们结合在一起，编译器的外观可方便大多数程序员使用，同时对少数懂得如何使用底层功能的人，它并不隐藏这些功能，如下图所示。



结构图

