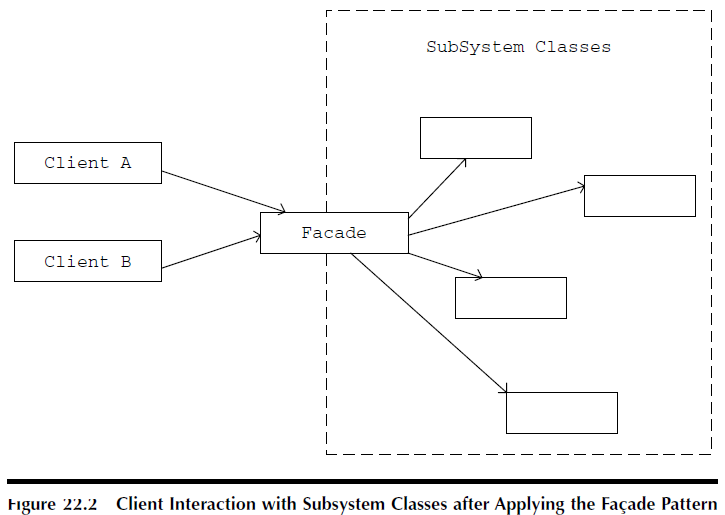
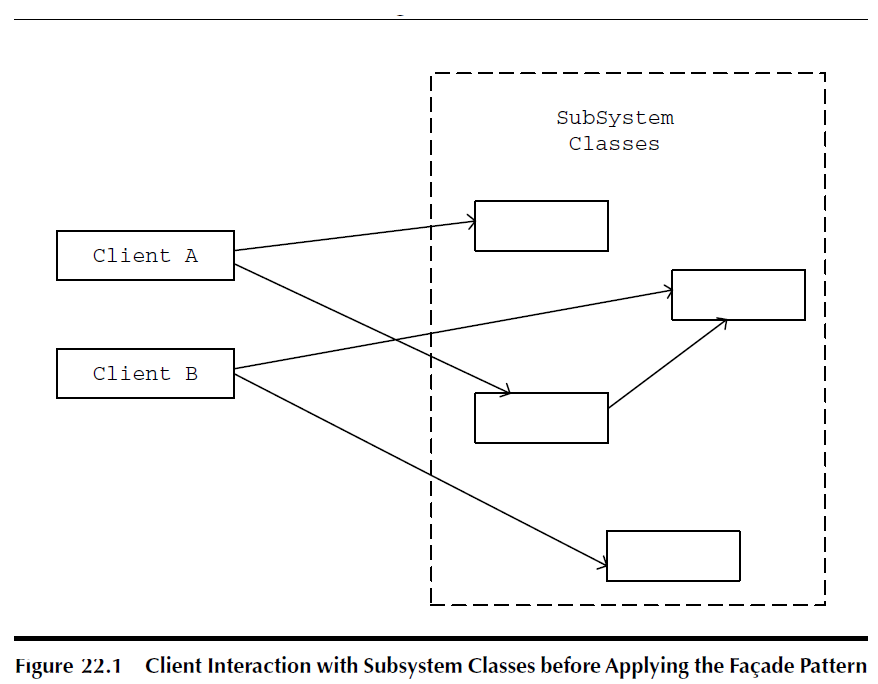
# El patrón de fachada

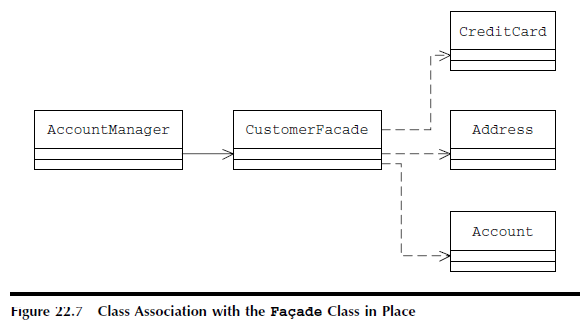
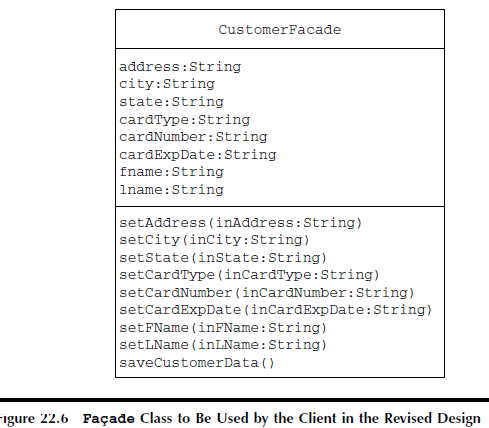
El patrón de fachada trata con un subsistema de clases. Un subsistema es un conjunto de clases que funcionan conjuntamente entre sí con el fin de proporcionar una funcionalidad. Por ejemplo, una clase de cuenta, una clase de dirección y una clase de tarjeta de crédito que trabajan juntas, como parte de un subsistema, proporcionan características de un cliente en línea.

* Los clientes de un subsistema pueden necesitar interactuar con varias clases de subsistemas para sus necesidades. Esta interacción directa conduce a un alto grado de acoplamiento entre los objetos de cliente y el subsistema (Figura 22.1).
* Cada vez que una clase de subsistema sufre un cambio, como un cambio en su interfaz, todas sus clases de clientes dependientes pueden verse afectadas.



El patrón de fachada es útil en tales situaciones dado que proporciona una interfaz simplificada de alto nivel para un subsistema que resulta en una menor complejidad y dependencia. Esto a su vez hace que el uso del subsistema sea más fácil y más manejable.

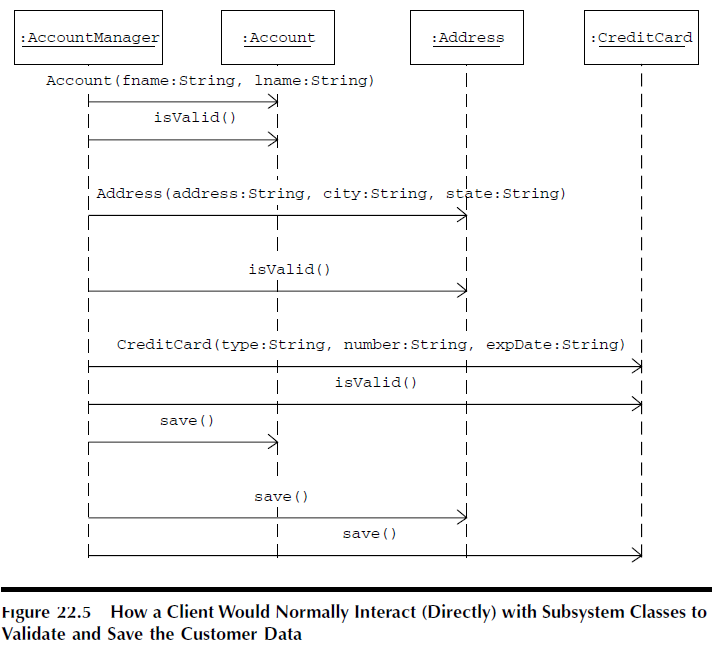
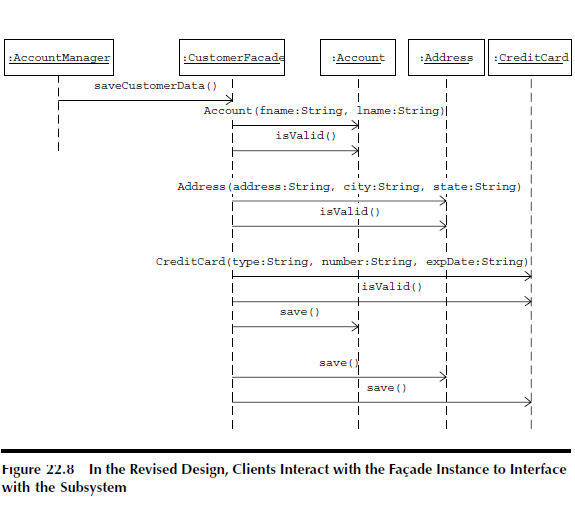
* Una fachada es una clase que proporciona una interfaz simplificada para que los clientes utilicen un subsistema.
* Los clientes interactúan con el objeto Fachada en lugar de interactuar directamente con las clases del subsistema.
* El objeto fachada asume la responsabilidad de interactuar con las clases del subsistema. Ineficaz
* El patrón fachada permite un acoplamiento débil entre un subsistema y sus clientes (Figure 22.2).

Consideraciones:

* El patrón fachada no esta diseñado para proporcionar funcionalidad adicional.
* Nunca se retorna un componente del subsistema desde la fachada, es decir, utilizar un método como el que sigue: CreditCard getCreditCard()ya que expone los subsistemas internos al cliente y no se aprovechan los beneficios que propone el patrón fachada.
* El objetivo de una fachada es proporcionar una interfaz de alto nivel y, preferiblemente, un método típico de fachada debería ofrecer un servicio de negocio de alto nivel en lugar de realizar una tarea individual de bajo nivel.

El diagrama de secuencia en la Figura 22.5 representa el flujo de mensajes entre objetos. La aplicación del patrón Fachada en este caso puede conducir a un mejor diseño ya que promueve un bajo acoplamiento entre el cliente y los componentes del subsistema (clases de Dirección, Cuenta y Tarjeta de crédito en este caso).

Aplicando el patrón Fachada, definamos una clase Fachada CustomerFacade (Figura 22.6 y Listado 22.5) que ofrece una interfaz simplificada de nivel superior para el subsistema que consiste en clases de procesamiento de datos del cliente (Dirección, Cuenta y Tarjeta de crédito).