Adapter

Propósito

Convierte la interface de una clase en la interface que espera el cliente. El Adapter permite trabajar juntas a unas clases que tienen interfaces incompatibles.

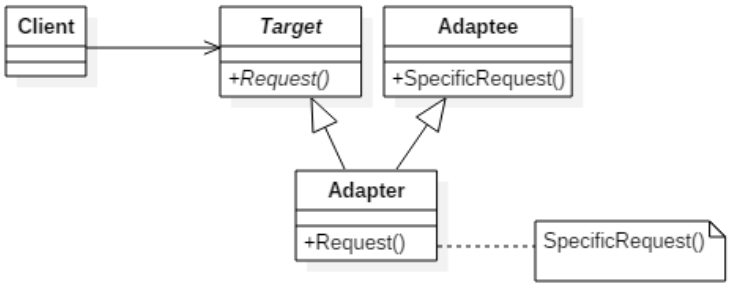
Aplicabilidad

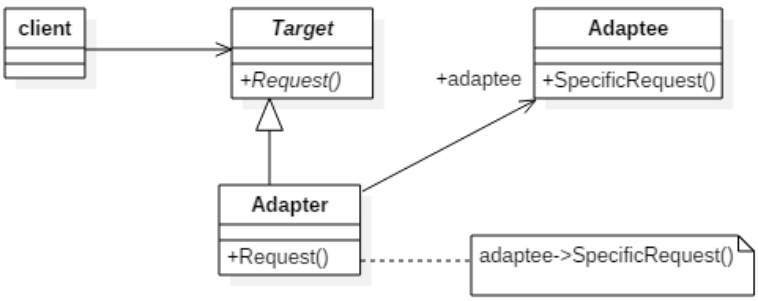
El Adapter se utiliza cuando:

- Se desea utilizar una clase existente pero su interface no encaja con la que necesitas
- Se desea crear una clase reutilizable que coopere con clases no relacionadas o incluso desconocidas, esto es, clases que no tienen necesariamente interfaces compatibles
- (adapter de objetos) Se necesita utilizar varias subclases pero no es práctico adaptar sus interfaces mediante la subclasificación. Un objeto adapter puede adaptar la interface de la clase padre.

Estructura

El primer diagrama utiliza herencia múltiple y el segundo lo resuelve mediante composición (Adapter "tiene-un" Adaptee).





Participantes

- Target: define la interface que utiliza el Client
- Client: colabora con los diferentes objetos mediante Target
- Adaptee: define una interface existente que necesita ser adaptada
- Adapter: adapta la interface de Adaptee a la interface de Target

Colaboraciones

- Los clientes invocan operaciones de la instancia Adapter, la cual invoca operaciones de Adaptee que responden a la petición

Consecuencias

Las dos propuestas del Adapter tienen diferentes compromisos. El Adapter basado en clases:

- Adapta Adaptee a Target en una clase concreta Adapter. Como consecuencia, un adapter de clases no funcionará bien cuando queramos adaptar una clase y todas sus subclases
- Permite que el Adapter sobreescriba algún comportamiento del Adaptee
- Lo resuelve mediante un único objeto, no hace falta ninguna otra indirección para obtener el Adaptee

En cuanto al adapter basado en la composición:

- Permite que un único Adapter funcione con muchos Adaptee (la propia clase y todas sus subclases)
- Hace más difícil sobreescribir comportamiento del Adaptee. Requeriría subclasificar el Adaptee y hacer que el Adapter se refiera a la subclase en vez del Adaptee.

Otros asuntos a considerar cuando se utilice el patrón Adapter:

1. ¿Cuánta adaptación realiza el Adapter?: Los adapters varían en cuanto al trabajo que requiere adaptar el Adaptee a Target. Hay un espectro de posibles trabajos, desde la simple conversión de la interface -cambio de nombres-, hasta soportar un conjunto diferente de operaciones. La cantidad de trabajo a realizar depende de cuánto de similar sean las interfaces de Target y Adaptee.
2. Pluggable adapters (adapters enchufables). Una clase es más reutilizable cuando minimizas las suposiciones que otras clases deben hacer para usarla. La adaptación de la interface nos permite incorporar nuestra clase en otros sistemas existentes que podrían esperar una interface diferente de la clase.
3. Utilizar adapters de dos direcciones para proporcionar transparencia. Esto permite que los clientes puedan usar cualquiera de las dos interfaces (Target o Adaptee), porque ambas son soportadas.