Patrón de diseño Adaptador

Es útil cuando hay una funcionalidad ya implementada en una clase que se puede proveer a un cliente, pero la firma de las operaciones no coincide con la del cliente.

* **Enfoque adaptador de Objeto:** Se utiliza una asociación en el objeto adaptador (wraper) al objeto adaptado, de esta manera se tiene acceso al método adaptado a través de esa referencia y desde el adaptador se provee la firmas compatibles de las operaciones requeridas y se utiliza la funcionalidad existente a través de la referencia al objeto adaptado.
* **Enfoque adaptador de clase:** El objeto adaptador es una especialización del objeto adaptado, de esta manera se heredan las funcionalidades de este para poder adaptar la firma de las operaciones y reutilizar la funcionalidad.

|  |  |
| --- | --- |
| Adaptador de Clase | Adaptador de Objeto |
| Basado en herencia | Basado en composición (wraper) |
| Solo se puede utilizar para adaptar la interfaz del adaptado. No se pueden adaptar las interfaces de sus subclases, ya que el adaptador está vinculado estáticamente con el adaptado cuando se crea. | Se puede utilizar para adaptar la interfaz del adaptado y todas sus subclases. |
| Debido a que el adaptador está diseñado como una subclase del adaptado, es posible sobrescribir parte del comportamiento del adaptado.  Nota: en Java, una subclase no puede sobrescribir un método que se declara como final en su clase principal. | No se pueden sobrescribir los métodos adaptados. Nota: Literalmente, no puede " sobrescribir " simplemente porque no hay herencia. Pero las funciones de contenedor “wraper”proporcionadas por el adaptador pueden cambiar el comportamiento según sea necesario. |
| El cliente tendrá cierto conocimiento de la interfaz del adaptado “adaptee” ya que la interfaz pública completa del adaptado es visible para el cliente. | El cliente y el adaptado están completamente desacoplados. Solo el adaptador conoce la interfaz del adaptado. |
| En aplicaciones Java:  Adecuado cuando la interfaz esperada está disponible en forma de una interfaz Java y no como una clase abstracta o concreta. Esto se debe a que el lenguaje de programación Java solo permite una única herencia.  Dado que un adaptador de clase está diseñado como una subclase de la clase adaptada, tampoco podrá subclasificar la clase de interfaz (que representa la interfaz esperada), si la interfaz esperada está disponible en forma de una clase abstracta u concreta. | En aplicaciones Java:  Adecuado incluso cuando la interfaz que  el objeto del cliente espera está disponible en forma de una clase abstracta.  También se puede usar si la interfaz esperada está disponible en forma de una interfaz Java. O cuando es necesario adaptar la interfaz  del adaptado y también todas sus subclases. |
| En aplicaciones Java:  Puede adaptar métodos con especificador de acceso protegido. | En aplicaciones Java:  No se pueden adaptar métodos con un especificador de acceso protegido, a menos que el adaptador y el adaptado estén diseñados para formar parte del mismo paquete. |
|  |  |



