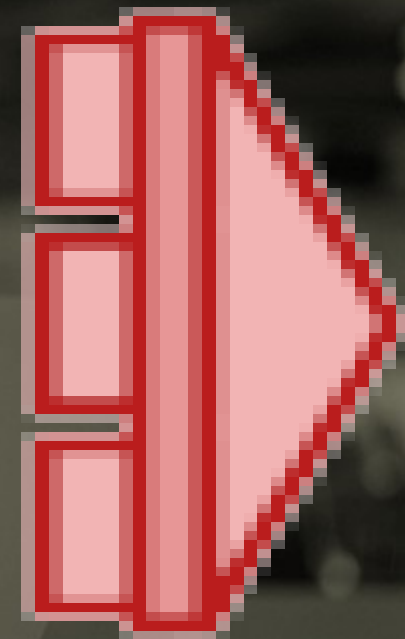
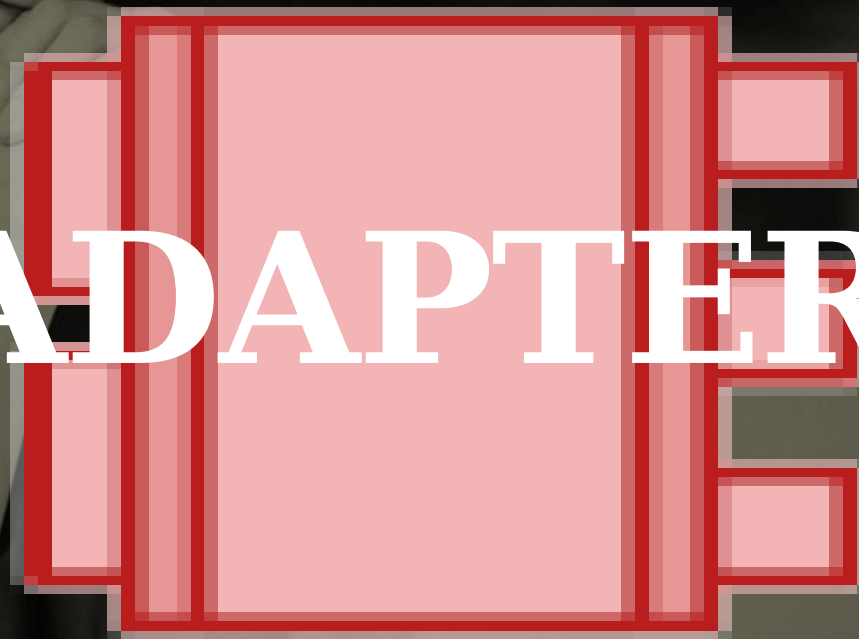
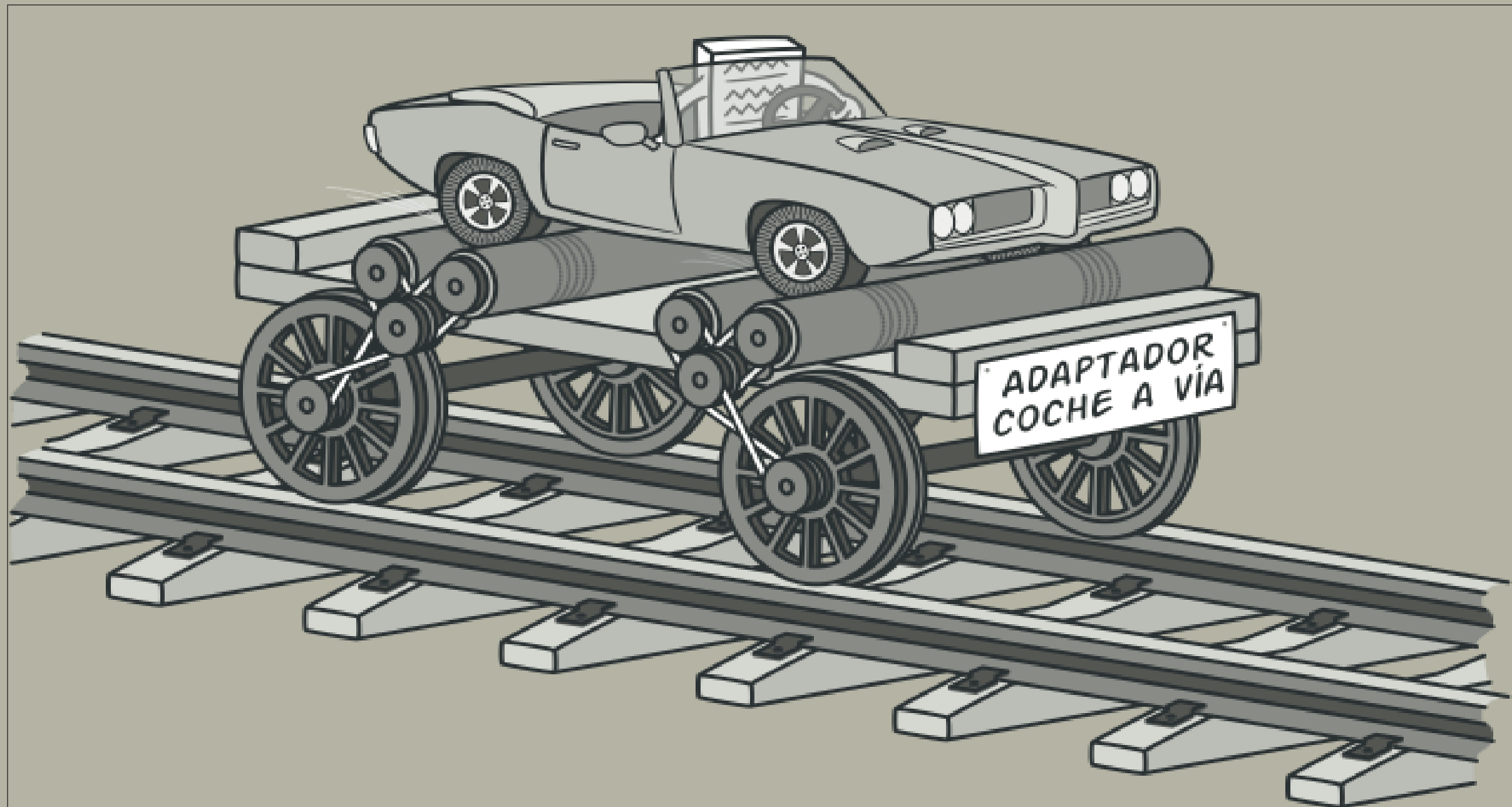




ADAPTER



patrón estructural
Adaptador, Envoltorio, Wrapper

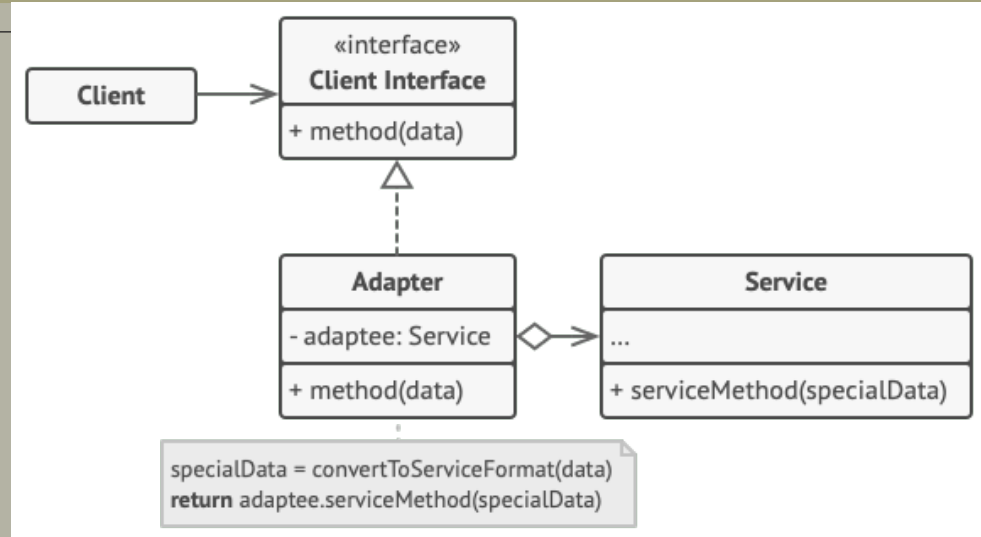


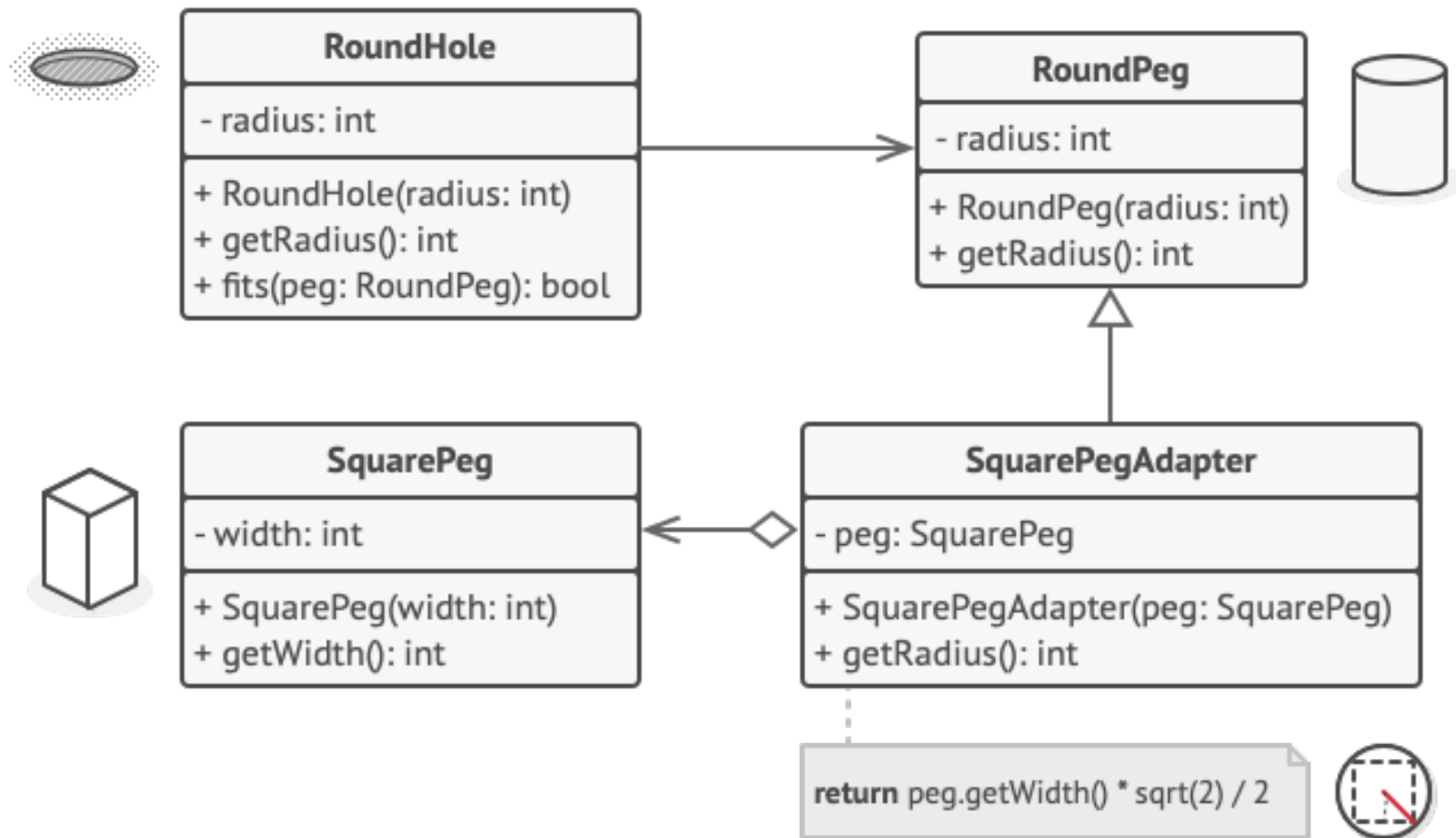
Problema o intención

- Se utiliza cuando necesitamos hacer compatibles dos interfaces, que de inicio no lo son. Muy útil cuando trabajamos con librerías externas.
- Ejemplo del mundo real: un adaptador para utilizar un conector español en un enchufe americano.

Estructura

1. La clase Cliente contiene la lógica de negocio existente del programa.
2. La Interfaz con el Cliente describe un protocolo que otras clases deben seguir para poder colaborar con el código cliente.
3. Servicio es alguna clase útil (normalmente de una tercera parte o heredada). El cliente no puede utilizar directamente esta clase porque tiene una interfaz incompatible.
4. La clase Adaptadora es capaz de trabajar tanto con la clase cliente como con la clase de servicio: implementa la interfaz con el cliente, mientras envuelve el objeto de la clase de servicio. La clase adaptadora recibe llamadas del cliente a través de la interfaz adaptadora y las traduce en llamadas al objeto envuelto de la clase de servicio, pero en un formato que pueda comprender.
5. El código cliente no se acopla a la clase adaptadora concreta siempre y cuando funcione con la clase adaptadora a través de la interfaz con el cliente. Gracias a esto, puedes introducir nuevos tipos de adaptadores en el programa sin descomponer el código cliente existente. Esto puede resultar útil cuando la interfaz de la clase de servicio se cambia o sustituye, ya que puedes crear una nueva clase adaptadora sin cambiar el código cliente.





Aplicación o usos conocidos

Este ejemplo del patrón **Adapter** se basa en el clásico conflicto entre piezas cuadradas y agujeros redondos.

El patrón Adapter finge ser una pieza redonda con un radio igual a la mitad del diámetro del cuadrado (en otras palabras, el radio del círculo más pequeño en el que quepa la pieza cuadrada)

Ventajas y desventaja

- **Ventajas:** hace que dos interfaces incompatibles, sean compatibles. Puede servir para encapsular clases que no controlamos, y que pueden cambiar.
-
- **Desventajas:** como muchos patrones, añade complejidad al diseño. Hay quién dice que este patrón es un parche, utilizado en malos diseños.
-

A black and white photograph of a person's hands clapping. The person is wearing a plaid shirt. In the foreground, there is a wooden desk with an open notebook and a smartphone lying on it. In the background, a laptop is visible. A dark grey rectangular box with a thin white border is overlaid on the left side of the image, containing the text '¡GRACIAS!'.

¡GRACIAS!