



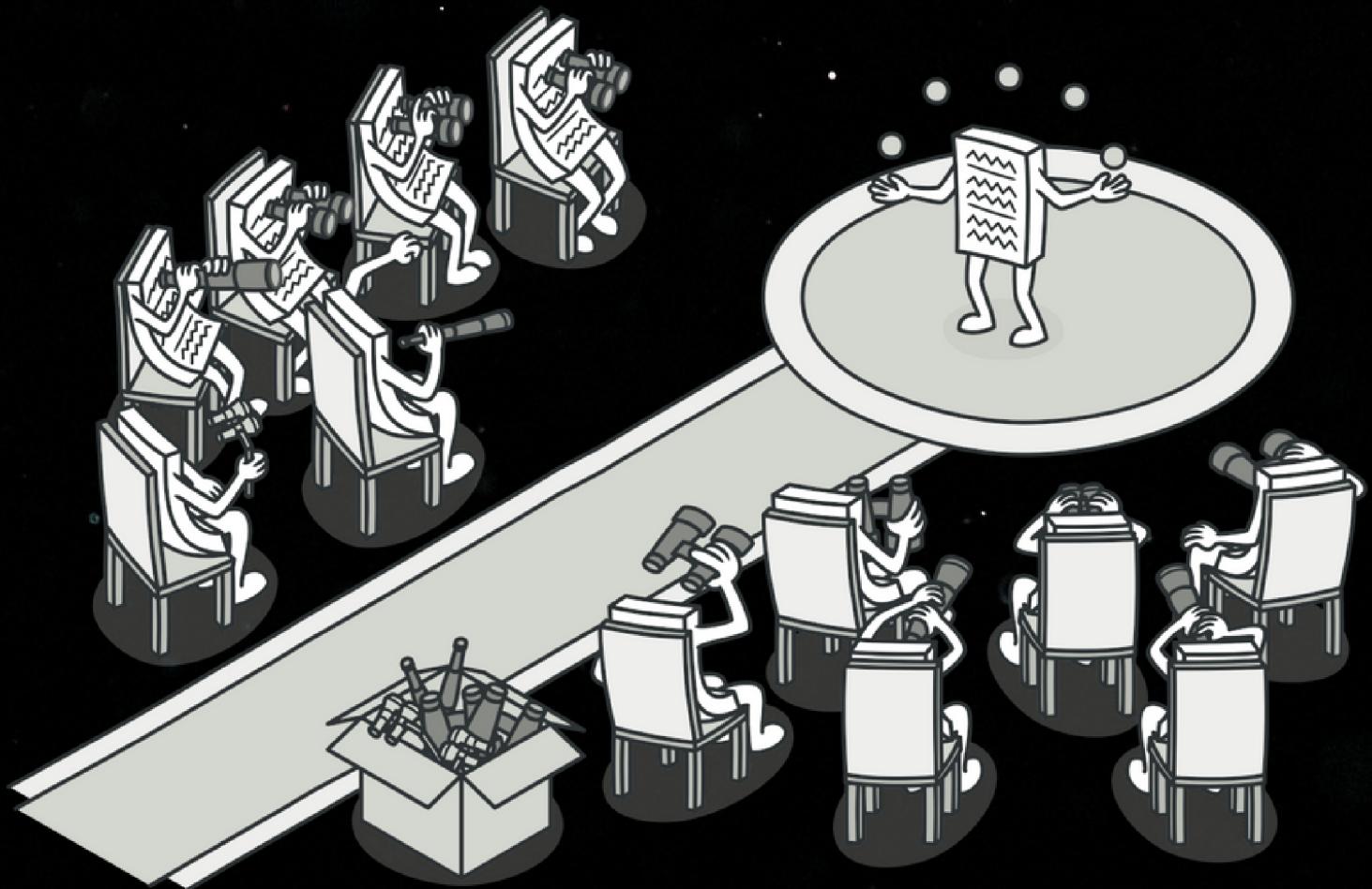
PATRÓN DE DISEÑO: OBSERVER

ALBA VELASCO

¿Qué es?

El patrón Observer es un patrón de diseño de comportamiento que te permite establecer una relación de uno a muchos entre objetos.

Cuando un objeto (sujeto) cambia su estado, todos los objetos dependientes (observadores).son notificados automáticamente y actualizados.



SUJETO OBSERVADO / SUBJECT

Es el objeto que es observado y mantiene una lista de todos los observadores registrados. Proporciona métodos para agregar y eliminar observadores, así como para notificar a los observadores sobre cambios en su estado.

OBSERVADOR / OBSERVER

Representa la interfaz que define el método de actualización que debe implementar cada observador. Los observadores se registran con el sujeto observado y serán notificados cuando ocurran cambios en el sujeto.

OBSERVADOR CONCRETO / CONCRETE OBSERVER

Son las clases que implementan la interfaz de Observador. Cada observador concreto mantiene una referencia al sujeto observado y, cuando se notifica de un cambio, actualiza su estado en función de la información proporcionada por el sujeto.

Aplicabilidad

Se podría utilizar el patrón Observer cuando los cambios en el estado de un objeto puedan necesitar cambiar otros objetos y el grupo de objetos sea desconocido de antemano o cambie dinámicamente.

Puedes utilizarlo cuando algunos objetos de tu aplicación deban observar a otros, pero sólo durante un tiempo limitado o en casos específicos.

Ejemplo, una lista de suscripción...por lo que los suscriptores pueden unirse o abandonar la lista cuando lo deseen.

VENTAJAS

Desacoplamiento

Escalabilidad

Notificaciones automáticas

Reusabilidad

INCONVENIENTES

Possible sobrecarga de notificaciones

Orden de notificación no garantizado

Dificultad en la depuración

¡VAMOS AL CÓDIGO!