IMAGE CAMPUS

Programación I

Relaciones entre Clases

Ignacio Mosconi

Intoducción

Existen distintos tipos de relaciones entre clases. Muchos de estos tipos de relaciones se pueden graficar mediante diagramas de clases.

Crear estos diagramas permite documentar el funcionamiento del *software*, así como facilitar el entendimiento del mismo. De esta manera, se vuelve más manejable y escalable.

Los tipos de relaciones más comunes son:

- Asociación
  - Agregación
    - Composición
- Dependencia
- Herencia

Asociación

Se trata de una relación estructural que especifica una conexión importante entre objetos.

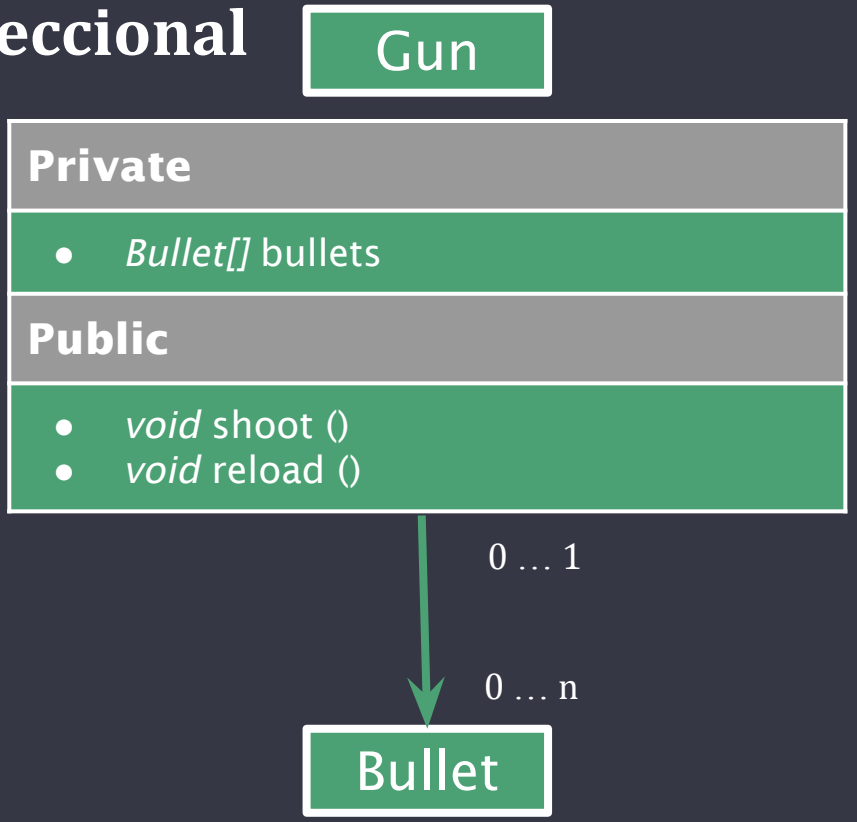
Las mismas disponen de una **navegación**; es decir, tienen el concepto de direccionalidad para representarse. Entonces, pueden ser **unidireccionales** o **bidireccionales**, dependiendo de las direcciones en las que se pueda dar la asociación.

Asimismo, pueden especificar una **multiplicidad**, la cual indica cuántas entidades pueden participar de la relación.

Ejemplo de Asociación Unidireccional

En un ejemplo de relación entre un arma y las balas, cada arma puede tener de 0 a n balas. Por su parte, las balas pertenecerán hasta a una sola arma como máximo.

La flecha nos indica el sentido de la relación. El arma es la que tiene las balas, y no al revés.



Ejemplo de Asociación Bidireccional

Suponiendo el ejemplo de un juego con zonas y jefes, podríamos pensar que cada zona tiene un (y solamente un) jefe. Luego, un mismo jefe puede ser el jefe de varias zonas.

En este caso, podemos hablar de una relación bidireccional.

Agregación

Consiste en un tipo de asociación más específica, en la que una entidad se “agrega” a otra.

Lo importante es que esta relación no es necesaria para que la entidad a la que se le agregan otras funcione como tal. Son como “accesorios” que una entidad puede o no tener.

Por ejemplo, un rifle puede o no tener una mira. No obstante, en cualquier caso va a poder ejecutar su función de disparar.

Ejemplo de Agregación

Suponiendo un ejemplo de una armadura a la que se le pueden encastrar runas para darle habilidades, la armadura en sí no necesita de runas para ser equipada por un personaje.

Se dice que las runas se “agregan” a la armadura, pero pueden usarse sin ellas.

En un diagrama, se representan con un “diamante” vacío del lado de la clase a la cual se le agregan objetos de otras.

Composición

Es considerada como un tipo más específico de agregación, una en la que la relación es estricta.

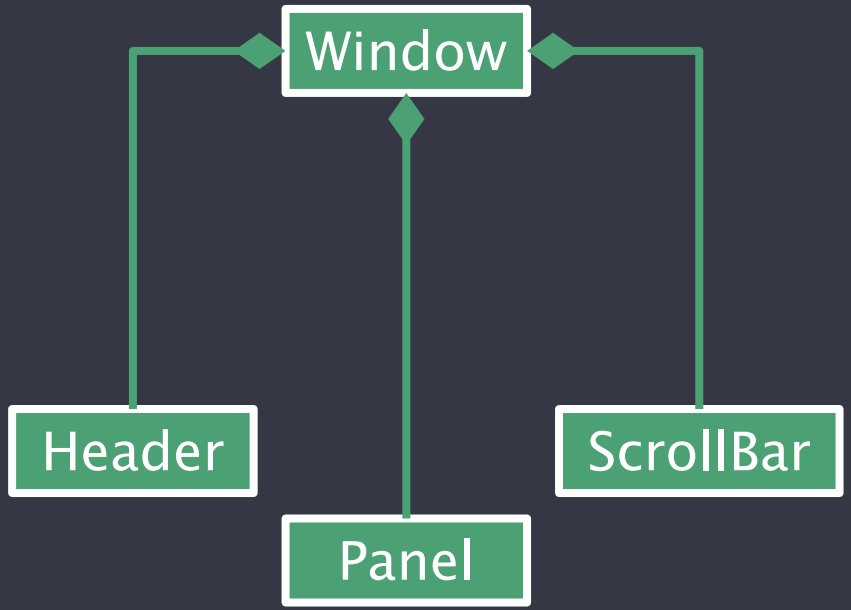
Esto implica que una clase está compuesta por objetos de otra clase, por lo que no puede funcionar como tal si los demás objetos no están.

Por ejemplo, un auto siempre requiere de ruedas para funcionar. Sin esas ruedas, no puede cumplir su función como vehículo.

Ejemplo de Composición

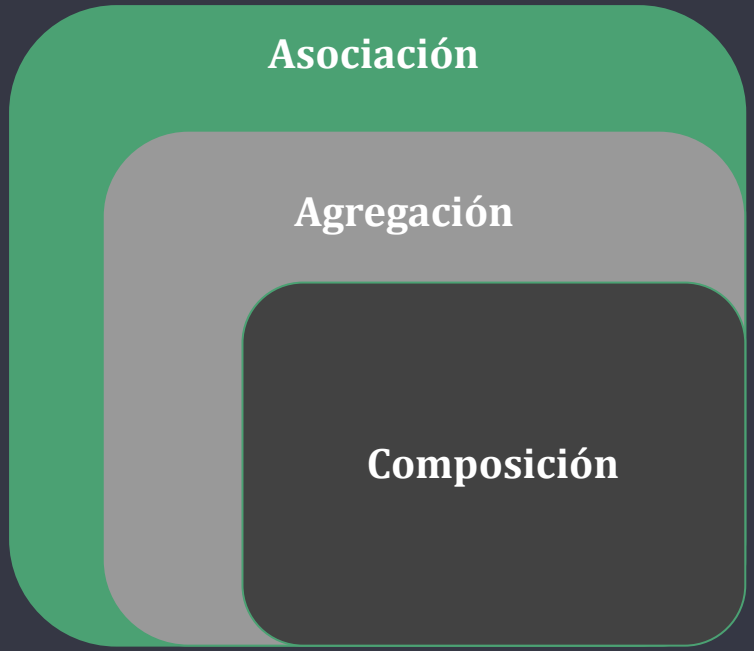
Si pensamos en el ejemplo de la ventana de una interfaz de usuario, podemos asumir que siempre necesitará de un encabezado, un panel y una barra de desplazamiento.

En este caso, si todas las entidades componen a la clase, entonces se usa un “diamante” lleno para indicar que las relación es estricta.



Tipos de Asociaciones

Todos los tipos de relaciones previamente introducidos se trataban de versiones más específicas de la **asociación**. De esta manera, una **agregación** es un tipo de asociación más específica, y la **composición** es un tipo de agregación más específica.



Dependencia

Las dependencias son consideradas como relaciones más débiles entre clases.

En ellas, se puede ver una relación de tipo “cliente” y “servidor”, donde la interacción se limita a consumir un servicio del servidor para ser usado en el cliente.

Por ejemplo, podemos pensar que una impresora que imprime documentos requiere que se le pase como parámetro el documento a imprimir. Sin esa información, no puede realizar una funcionalidad precisa. No obstante, el documento no se asocia, ni se agrega, ni se compone a la impresora. Se trata de una relación más débil.

Ejemplo de Dependencia