ATRIBUTOS

* + Publico: Accesible de todos lados.
* - Privado: Solo se podrá acceder a ellos dentro de la clase a través de los métodos de la misma clase.
* # protegido: no será accesible fuera de clase pero si por los métodos de la clase y por aquellos métodos de subclases heredadas.

INTERFACES

* Agregan una funcionalidad a una clase.
* A través de una especie de contrato.
* Es la acción que se cumple, da igual como lo haga, pero deben cumplir el objetivo, por ejemplo doblar a la izquierda en un auto, uno gira el volante y dobla pero no sabe el proceso interior que se produjeron para que girando el manubrio girara el auto.
* Lista de métodos sin su cuerpo, las clases hijas le darán ese cuerpo.
* No se pueden crear objetos a través de interfaces.

RELACIONES

<https://www.youtube.com/watch?v=BmnY7oEN9hQ>

* **Dependencia**: corresponde a una relación entre clases en la que la modificación de una puede afectar a la otra sin que necesariamente sea esto inverso. Pasar un objeto por parámetro.
* **Asociación :** Se utiliza para describir que dos clases trabajan juntas para lograr un objetivo
* **Agregación**(usa) (tiene un) la pertenencia es más débil, “un cliente compra productos”. pueden existir sin que al eliminar uno desaparezca el otro o viceversa, flota a barco de guerra. Profesor tiene clases, ambos tanto profesor como clase siguen existiendo independientemente. Se grafica con un rombo transparente o blanco.
* **Composición(posee):** Diamante o rombo negro. Tienen el mismo ciclo de vida “un pedido tiene productos”. Una clase contiene a otra clase
* **Herencia o generalización:**Se designa por una flecha con la punta blanca o hueca

Se destina a la clase base o padre

Acoplamiento como los cambios de unas clases afectan a los cambios de otras clases.