**Relaciones entre clases**

* Clientela
* Composición ("tiene un")
* Anidamiento
* Herencia ("es un" / extends)

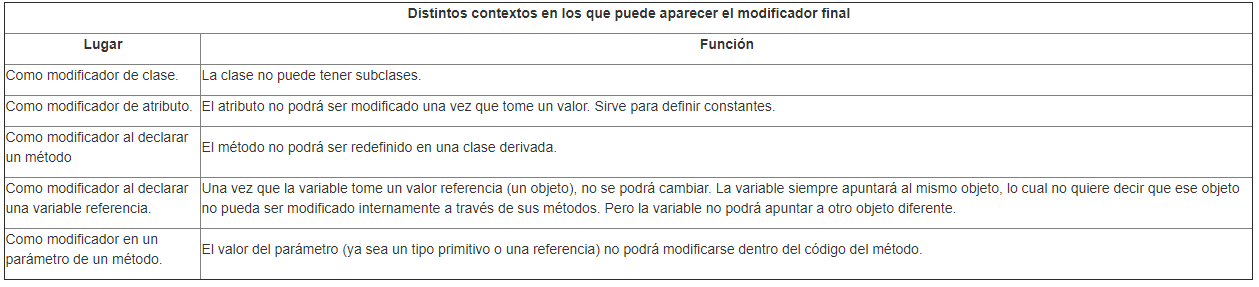
[Constructores / constructor copia]

[Referencia super / @Override]

[Jeraquía de clases con Object como clase base para todas]

[Modificador "abstract" > clases abstractas no instanciables (pueden contener métodos implementados)]

[Interfaces / "interface" en lugar de "class" / se usan con implements / sin métodos implementados / una clase puede implementar varias (a diferencia de las clases abstractas)]



Polimorfismo

[Early (static) binding > compilation time / late binding (ligadura dinámica) > run time]

**Se realiza una conversión implícita o automática** de **subclase** a **superclase** siempre que sea necesario.

La conversión de **superclase** a **subclase** debe hacerse de forma **explícita** (y puede producir excepciones por falta de info)**.**