Virrey Morcillo
Instituto de Educación Secundaria

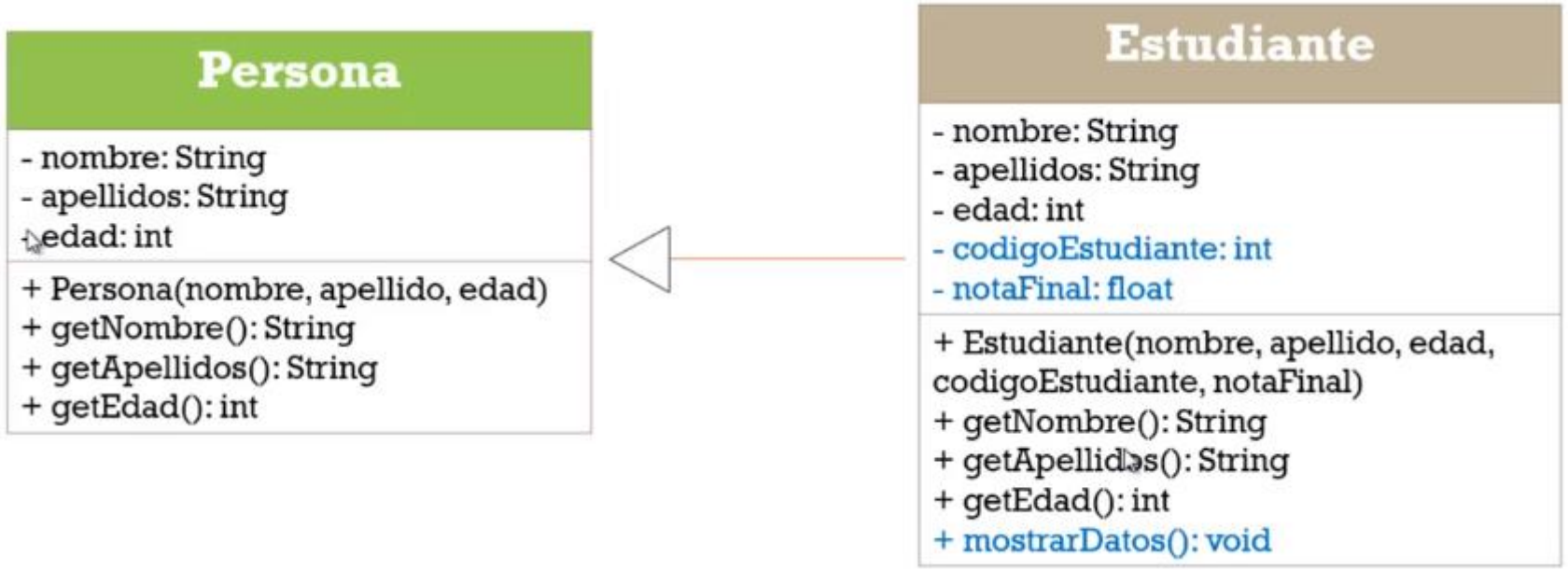
# UT 1. POO, EDA y API JAVA.

Herencia

Virrey Morcillo
Instituto de Educación Secundaria

# Herencia

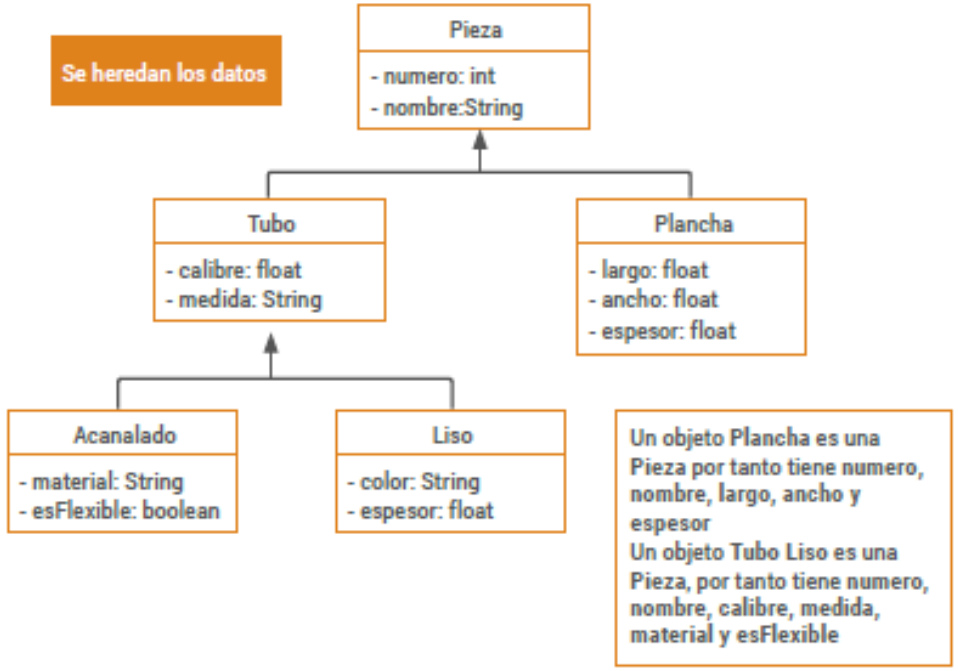
- ¿Qué es la herencia?
  - La herencia es una forma de reutilización de software en la que se crea una nueva clase al absorber los miembros de una ya existente.

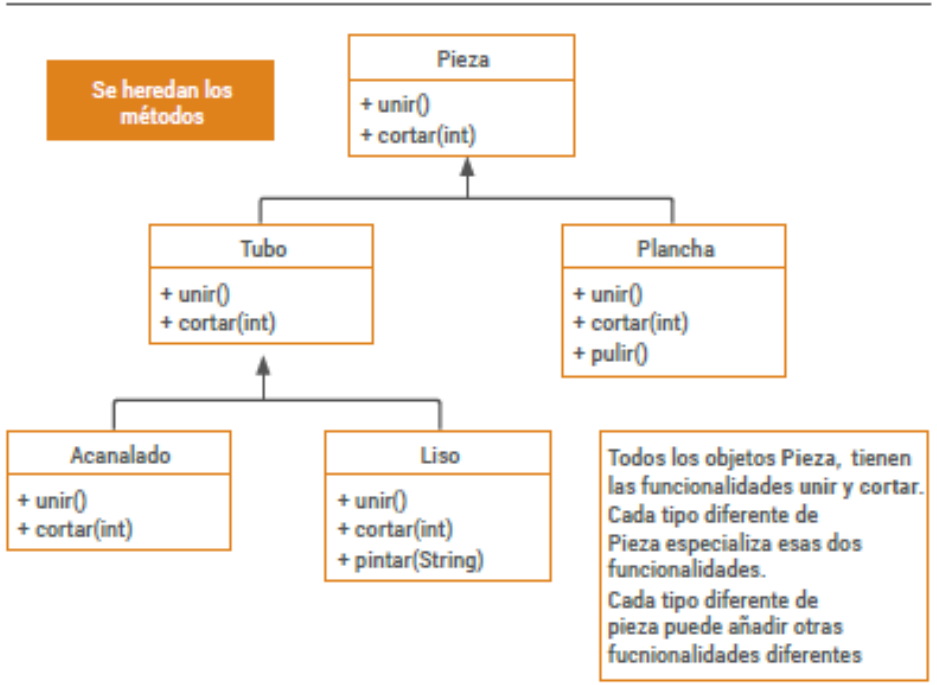


Virrey Morcillo
Instituto de Educación Secundaria

# Herencia

Las subclases heredan las variables miembro de la superclase
Las subclase heredan las funciones miembro de la superclase, respetando el tipo de acceso.



Virrey Morcillo
Instituto de Educación Secundaria

# Constructores y herencia, super

- Al crear un objeto de una subclase, el objeto se tiene que construir como un objeto de la superclase.
- Esto lo implementamos usando la palabra super dentro de los constructores de la subclase.
- La forma de invocar a super tiene que estar de acuerdo con uno de los costructores de la superclase.
- Las funciones también se heredadan  cualquier objeto de una subclase podrá invocar una de las funcionalidades de la superclase.

Virrey Morcillo
Instituto de Educación Secundaria

# Sobreescritura de miembros

- Cuando necesitamos que los objetos de las subclases tengan funciones especializadas distintas de las heredadas se redefinen en las subclases.
- Para que exista “override” las funciones en la superclase y en las subclases tienen que tener exactamente la misma forma.
- Dentro de la función sobrescrita en la subclase, se pued invocar a cualquier otra de la superclase, usando la palabra super.

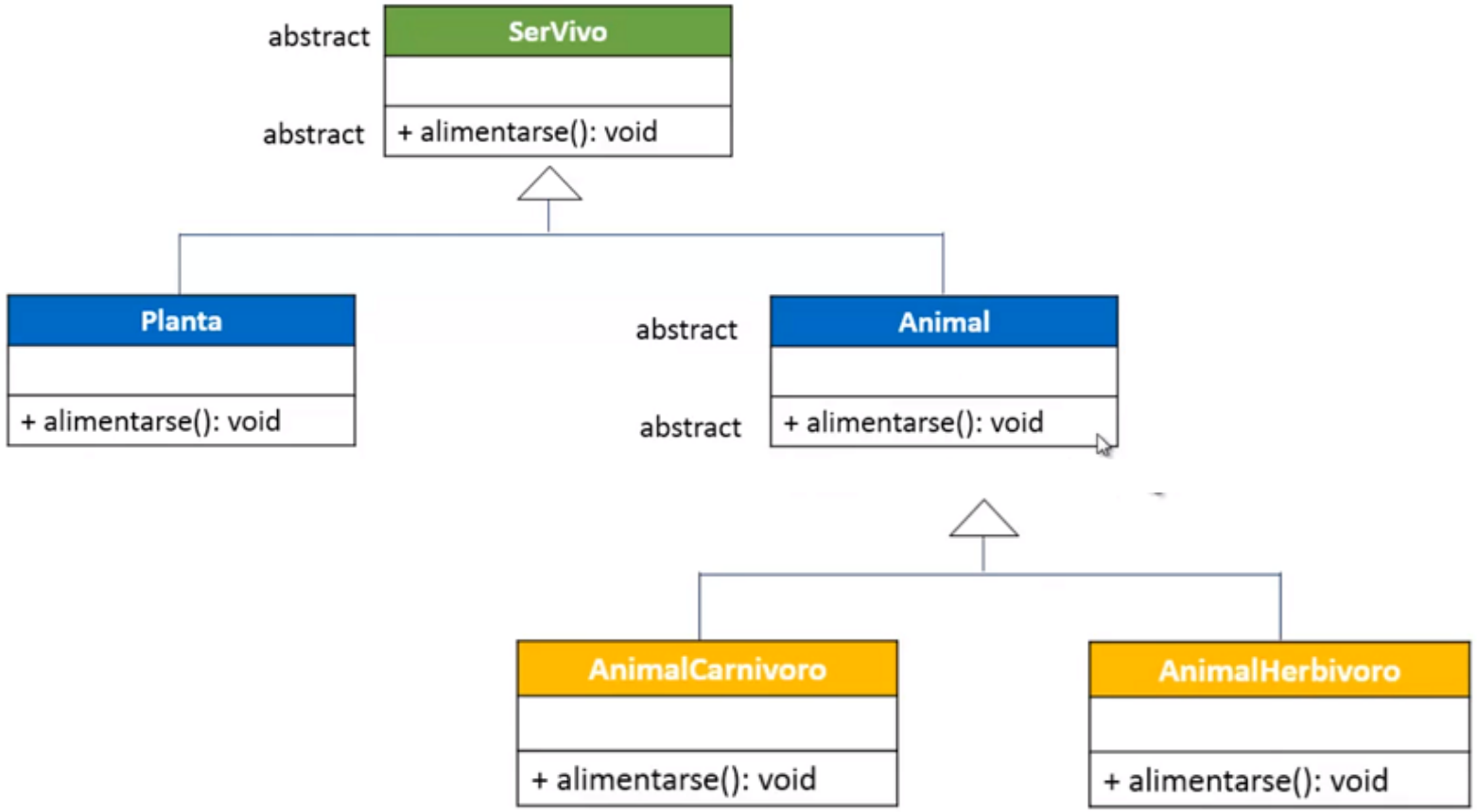
Instituto de Educación Secundaria Virrey Morcillo

# Clases y métodos abstractos

- Si de una clase nunca se van a instanciar objetos se denomina abstracta.
- Características:
  - Nunca se instancian objetos de esa clase.
  - Se utiliza solo como superclase.
  - Sirve para proporcionar una clase apropiada.
  - Los métodos abstractos deben ser implementados en las clases hijas.

Virrey Morcillo
Instituto de Educación Secundaria

# Clases y métodos abstractos



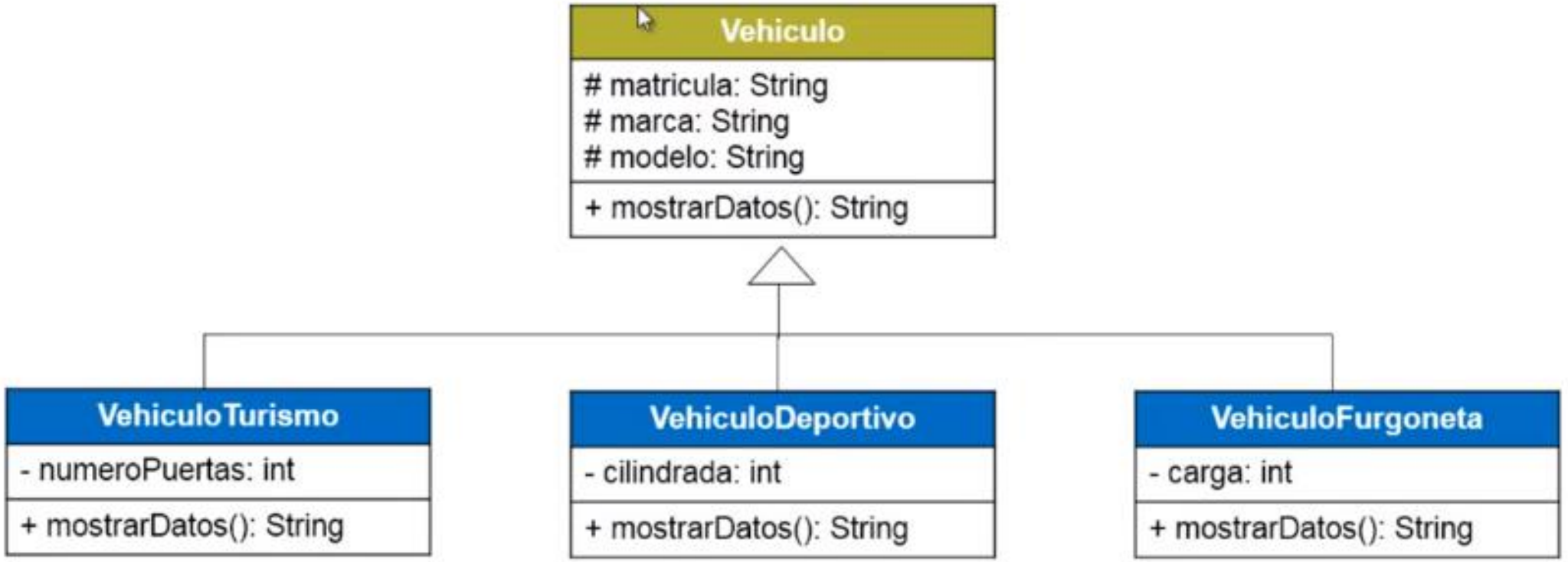
Virrey Morcillo
Instituto de Educación Secundaria

# Polimorfismo

- Es una relación de tipo herencia, un objeto de la superclase puede almacenar un objeto de cualquiera de sus subclases.
- Esto significa que la clase padre o superclase es compatible con los tipos que derivan de ella. Pero no al revés.
- Asociado a esto tendremos el upcasting.

Virrey Morcillo
Instituto de Educación Secundaria

# Polimorfismo



Virrey Morcillo
Instituto de Educación Secundaria

# Downcasting

- Conversión descendente de tipos.
  - Deseamos ir en la jerarquía de clases hacia abajo
  - Queremos convertir un objeto de clase superior en un objeto de clase inferior.
  - Para poder realizar esto previamente debe haber existido un upcasting.

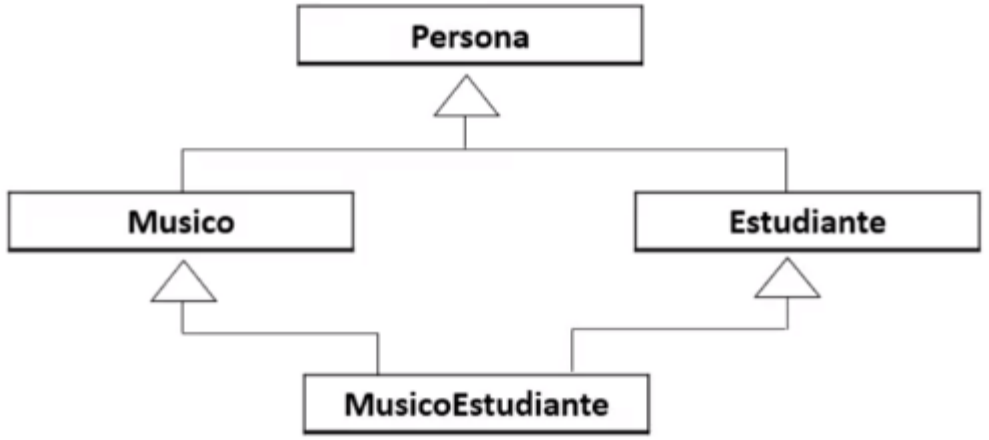
Clase_padre objeto1 = new Clase_hija(atributo, atributo2, **atributo3**);

Clase _hija objeto2 = (Clase_hija) objeto2;

Virrey Morcillo
Instituto de Educación Secundaria

# Interfaces

- Herencia Múltiple:
  - Herencia Múltiple hace referencia a la característica de los lenguajes de programación orientados a objetos en la que una clase puede heredar atributos y métodos de más de una superclase.



Virrey Morcillo
Instituto de Educación Secundaria

# Interfaces

- Interfaces en Java.
  - Permite simular la herencia múltiple.
  - La interfaz solo puede ser public o default.
  - Todos sus métodos son abstractos.
  - Todos sus atributos son final.
  - Usamos las palabras reservadas interface.
  - implements