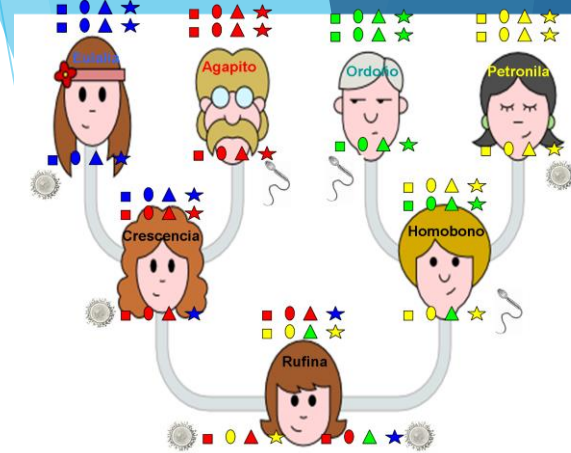


Contenido

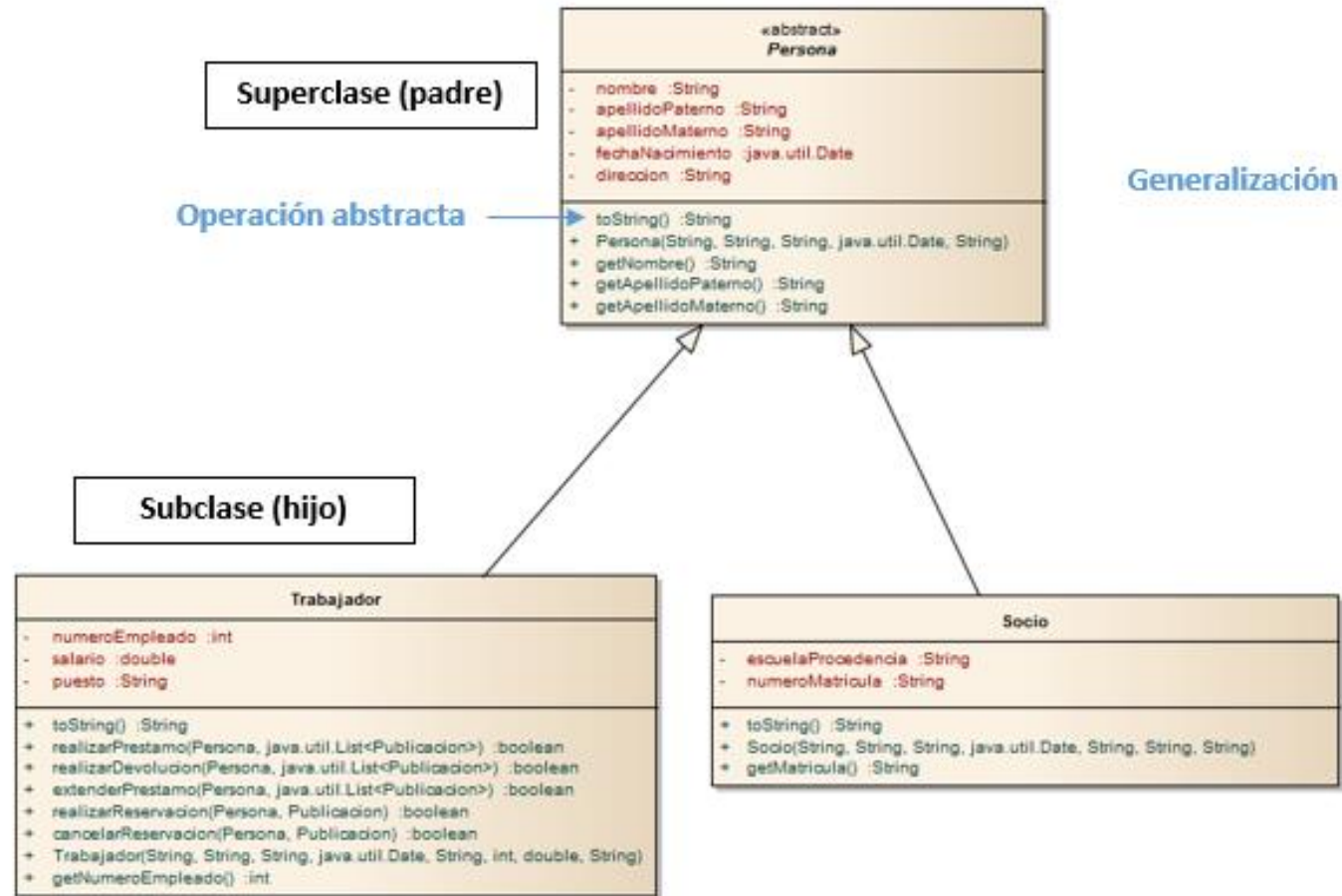
- ▶ Abstracción
- ▶ Encapsulación
- ▶ Herencia

Herencia (1/3)

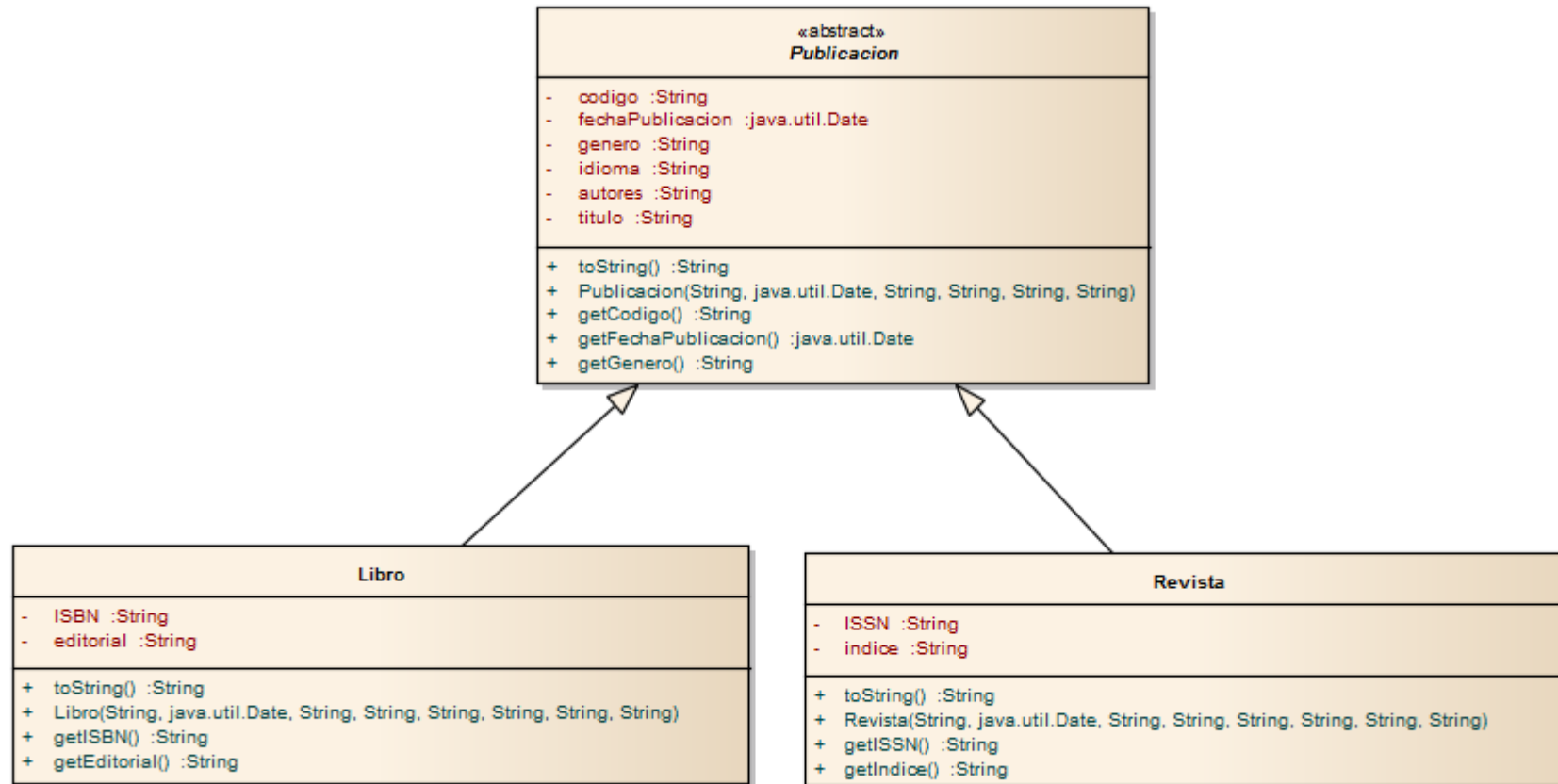
- ▶ La herencia es el proceso mediante el cual un objeto puede adquirir propiedades de otro
- ▶ La clase hereda las propiedades generales de su padre
- ▶ La herencia es el mecanismo que le permite a un objeto ser una instancia específica de una clase más general
- ▶ En este ámbito, se introducen los términos de *subclases* y *superclases*
- ▶ Las subclases contiene los atributos y métodos de la clase de la cual se deriva (superclase)
- ▶ La herencia es una potente abstracción para compartir similitudes entre clases
- ▶ Puede representarse visualmente de forma jerárquica, comenzando con la clase base llamada también superclase de la cual se derivan las clases secundarias



Herencia (2/3)



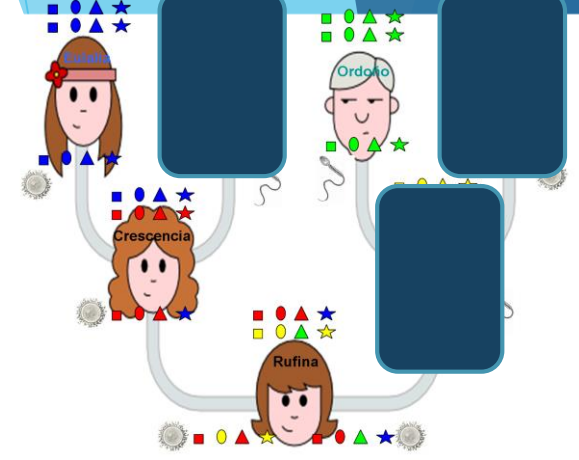
Herencia (3/3)



Implementación en Java

- ▶ Sobre-escritura de métodos
 - ▶ Consiste en cambiar el comportamiento de un método que tiene la misma firma en una clase derivada
- ▶ Si el método de la clase base se pretende que no sea sobre-escrito entonces el método debe ser declarado utilizando la palabra reservada **final**
- ▶ Ejemplo:
 - ▶ Si la clase Persona es la clase base (superclase, padre) y se busca tener una clase derivada entonces la herencia se realiza de la siguiente forma:
 - ▶ Superclase: `class Persona{....}`
 - ▶ Clase derivada (subclase): `class Trabajador extends Persona{....}`
 - ▶ Clase derivada (subclase): `class Socio extends Persona{....}`
 - ▶ Para acceder a elementos de la clase base se utiliza la palabra reservada **super**.

Herencia Múltiple



- ▶ En caso de que una clase tenga más de un padre, hereda de ambos
- ▶ Estas propiedades (atributos, operaciones) son la unión de los padres
- ▶ En lenguajes de programación como Java no es posible implementar la herencia múltiple