La herencia es un pilar importante de OOP (Programación Orientada a Objetos). Es el mecanismo en Java por el cual una clase permite heredar las características (atributos y métodos)

En el lenguaje de Java, una clase que se hereda se denomina **superclase(clase madre o clase padre**. La clase que hereda se llama **subclase(clase hija)**. Por lo tanto, una subclase es una versión especializada de una clase madre. Hereda todas las variables y métodos definidos por la clase madre y agrega sus propios elementos únicos.

* **Clase Madre**: la clase cuyas características se heredan se conoce como clase madre (o una clase base o una clase principal).
* **Subclase**: la clase que hereda la otra clase se conoce como subclase (o una clase derivada, clase extendida o clase hija). La subclase puede agregar sus propios campos y métodos además de los campos y métodos de la superclase.
* **Reutilización**: la herencia respalda el concepto de “reutilización”, es decir, cuando queremos crear una clase nueva y ya hay una clase que incluye parte del código que queremos, podemos derivar nuestra nueva clase de la clase existente. Al hacer esto, estamos reutilizando los campos/atributos y métodos de la clase existente.

Podemos definir la herencia como la capacidad de crear clases que adquieren de manera automática los miembros (atributos y métodos) de otras clases que ya existen, pudiendo al mismo tiempo añadir atributos y métodos propios.

La palabra clave utilizada para la herencia es **extends**

***extends***: Esta palabra reservada, indica a la clase hija cual va a ser su clase madre.

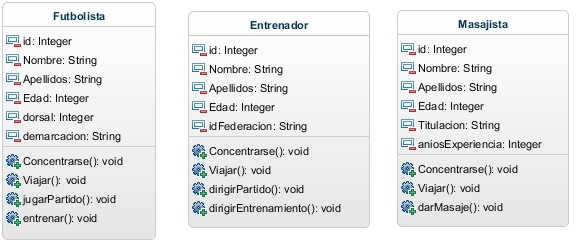
***super***: sirve para llamar al constructor de la clase madre.

Tenga en cuenta que durante la herencia **solo se crea(instancia) el objeto de la subclase**, no de la clase madre.

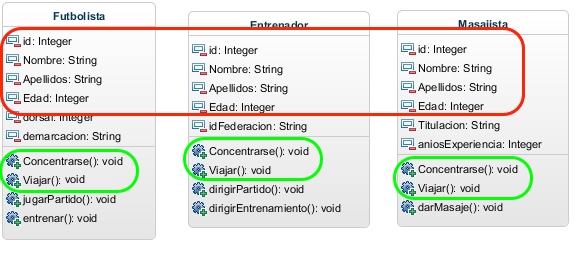
"La herencia es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente. La herencia permite compartir automáticamente métodos y datos entre clases, subclases y objetos."

De primeras, esta definición parece medio complicada, pero lo veremos con un ejemplo.

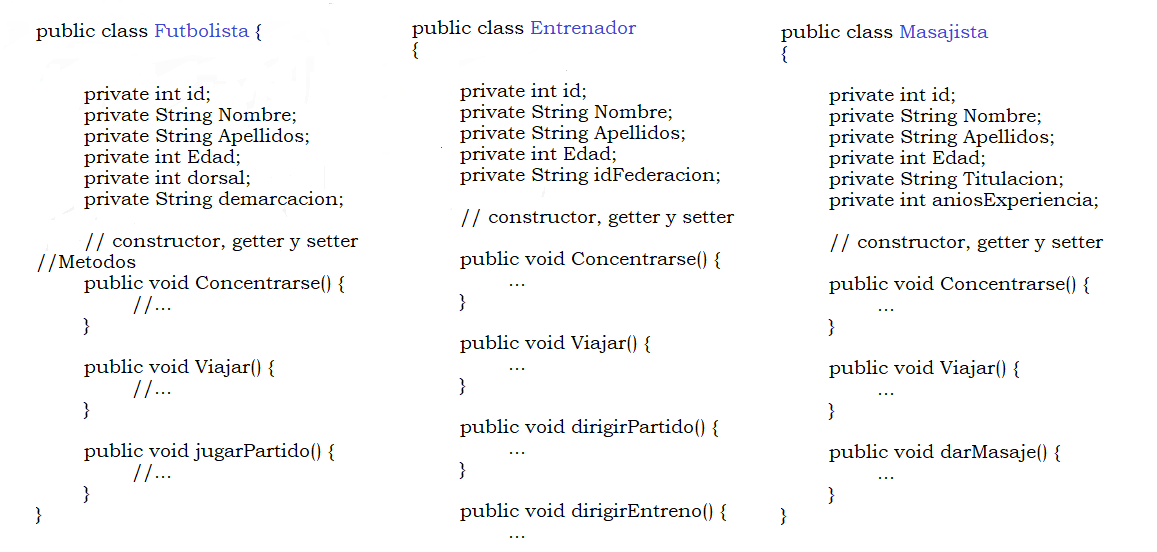
Supongamos que tenemos las siguientes clases:



Vemos que en las tres clases, tenemos atributos y métodos que son comunes a todas.

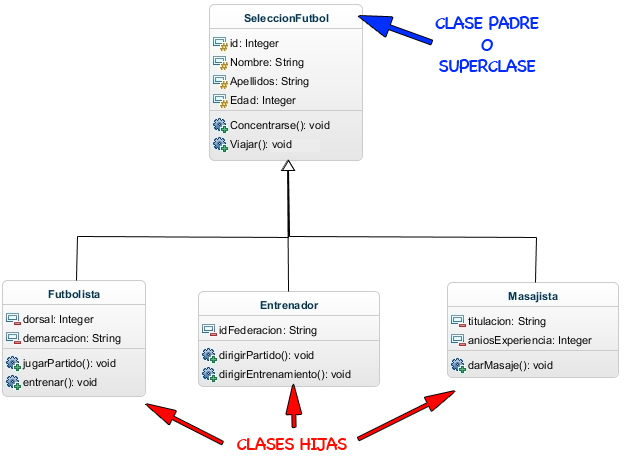


Si creamos las tres clases por separado, de manera normal, seria algo como esto:



Al ver esto nos damos cuenta que hay mucho código repetido en estas clases. Atributos y métodos que se repiten. Con la herencia podemos sacar el factor común entre todas ellas. Crearemos una clase con el código que es comuna las tres clases. Esta es la superclase o claseMadre, o clasePadre. En las clases hijas quedara solo el código que es específico de ellas. Las clases hijas, heredan de la clase madre, todos los atributos y métodos públicos, que esta tiene.

Con la herencia, nos quedaría una clase madre y tres clases hijas.



Con esto el código repetido desaparece y nos quedaría lo siguiente:

