

Enumerado (o Enum)

Un enumerado (o Enum) es una clase "especial" (tanto en Java como en otros lenguajes) ***que limitan la creación de objetos a los especificados explícitamente en la implementación de la clase.*** La única limitación que tienen los enumerados respecto a una clase normal es que, si tiene constructor, este debe de ser privado para que no se puedan crear nuevos objetos.

Vamos a empezar con un sencillo ejemplo sobre una clase Enum. Volviendo al ejemplo relacionados con el fútbol, tenemos que lo futbolistas están caracterizados por un puesto dentro de la cancha a la hora de jugar un partido de fútbol, por lo tanto, los puestos en los que puede jugar un futbolista son finitas y por tanto se pueden enumerar en: Arquero, Defensa, Centrocampista y Delantero. Con esta especificación podemos crearnos la siguiente clase "Enum" llamada "Puesto":

```
public enum Puesto
{
    ARQUERO, DEFENSA, CENTROCAMPISTA, DELANTERO
}
```

Por convenio (aunque lo podés poner como quieras) los nombres de los enumerados se escriben en mayúsculas.

Es muy importante entender que un "**Enum**" en java ***es realmente una clase*** (cuyos objetos solo pueden ser los definidos en esta clase: ARQUERO, ..., DELANTERO) ***que hereda*** de la clase "**Enum(java.lang.Enum)**" y por tanto los enumerados tienen una serie de métodos heredados de esa clase padre. A continuación, vamos a mostrar algunos de los métodos más utilizados de los enumerados:

```
public enum Puesto{ARQUERO, DEFENSA, CENTROCAMPISTA, DELANTERO}

Puesto delantero = Puesto.DELANTERO;    // Instancia de un enum de la clase puesto
delantero.name();    // Devuelve un String con el nombre de la constante (DELANTERO)
delantero.toString();    // Devuelve un String con el nombre de la constante (DELANTERO)
delantero.ordinal();    // Devuelve un entero con la posición del enum según está declarada (3).

delantero.compareTo(Enum otro);    // Compara el enum con el parámetro según el orden en el que están declarados lo enum

Puesto.values();    // Devuelve un array que contiene todos los enum
```

Visto cuales son los métodos más utilizados dentro de los enumerados, vamos a poner un ejemplo para ver los resultados que nos devuelven estos métodos. Dado el siguiente fragmento de código:

```
Puesto delantero = Puesto.DELANTERO;
Puesto defensa = Puesto.DEFENSA;

// Devuelve un String con el nombre de la constante
System.out.println("delantero.name()= "+delantero.name());
System.out.println("defensa.toString()= "+defensa.toString());

// Devuelve un entero con la posición de la constante según está declarada.
System.out.println("delantero.ordinal()= "+delantero.ordinal());

// Compara el enum con el parámetro según el orden en el que están declaradas las constantes.
System.out.println("delantero.compareTo(defensa)= "+delantero.compareTo(defensa));
System.out.println("delantero.compareTo(delantero)= "+delantero.compareTo(delantero));

// Recorre todas las constantes de la enumeración
for(Puesto d: Puesto.values()){
    System.out.println(d.toString()+" - ");
}
```

Tenemos como salida los siguientes resultados:

```
delantero.name()= DELANTERO
defensa.toString()= DEFENSA
delantero.ordinal()= 3
delantero.compareTo(defensa)= 2
delantero.compareTo(delantero)= 0
ARQUERO - DEFENSA - CENTROCAMPISTA - DELANTERO
```

Como ya se ha dicho un enum es una clase especial que limita la creación de objetos a los especificados en su clase (por eso su constructor es privado, como se ve en el siguiente fragmento de código); pero estos objetos pueden tener atributos como cualquier otra clase. En la siguiente declaración de la clase, vemos un ejemplo en donde definimos un enumerado "Equipo" que tiene dos atributos; el nombre y el puesto que obtuvo en la liga.

```
public enum Equipo
{
    BARÇA("FC Barcelona",1), REAL_MADRID("Real Madrid",2),
    SEVILLA("Sevilla FC",4), VILLAREAL("Villareal",7);
    private String nombreClub;
    private int puestoLiga;

    private Equipo (String nombreClub, int puestoLiga) {
        this.nombreClub = nombreClub;
        this.puestoLiga = puestoLiga;
    }
    public String getNombreClub() {
        return nombreClub;
    }
    public int getPuestoLiga() {
        return puestoLiga;
    }
}
```

Como se ve BARÇA, REAL_MADRID, etc. son el nombre del enumerado (u objetos de la clase Equipo) que tendrán como atributos el "nombreClub" y "puestoLiga". Como se ve en la clase **definimos un constructor que es *privado*** (es decir que solo es visible dentro de la clase Equipo) y solo definimos los métodos "get". Para trabajar con los atributos de estos enumerados se hace de la misma manera que con cualquier otro objeto; se instancia un objeto y se accede a los atributos con los métodos get. En el siguiente fragmento de código vamos a ver cómo trabajar con enumerados que tienen atributos:

```
// Instanciamos el enumerado
Equipo villareal = Equipo.VILLAREAL;

// Devuelve un String con el nombre de la constante
System.out.println("villareal.name()= "+villareal.name());

// Devuelve el contenido de los atributos
System.out.println("villareal.getNombreClub()= "+villareal.getNombreClub());
System.out.println("villareal.getPuestoLiga()= "+villareal.getPuestoLiga());
```

Como salida de este fragmento de código tenemos lo siguiente:

```
villareal.name()= VILLAREAL
villareal.getNombreClub()= Villareal
villareal.getPuestoLiga()= 7
```

Es muy importante que tengas claro que los enumerados no son Strings (aunque pueden serlo), sino que son objetos de una clase que solo son instanciables desde la clase que se implementa y que no se puede crear un objeto de esa clase desde cualquier otro lado que no sea dentro de esa clase. Es muy común (sobre todo cuando se está aprendiendo que son los enumerados) que se interprete que un enumerado es una lista finita de Strings y en realidad es una lista finita de objetos de una determinada clase con sus atributos, constructor y métodos getter aunque estos sean privados.