

# Anexo 2: Tipos Enumerados

## Contenido

1.	Tipos Enumerados .....	2
1.1.	Tipos enumerados simples .....	2

## 1. Tipos Enumerados

Además de las colecciones habituales de Java, JavaFX introduce nuevas: ObservableList, ObservableMap

### 1.1. Tipos enumerados simples

Un tipo enumerado define una lista de valores enumerados. Cada valor es un identificador. Por ejemplo la siguiente sentencia declara un tipo, llamado MiColorFavorito, con valores ROJO, AZUL, VERDE y AMARILLO en este orden.

```
enum MiColorFavorito {ROJO, AZUL, VERDE, AMARILLO};
```

Un valor de un tipo enumerado es como una constante y por lo tanto, por convención, se escribe con mayúsculas. Así, la declaración anterior utiliza RED, no rojo. Por convención, un tipo enumerado se nombra como una clase, con la primera letra de cada palabra en mayúscula.

Una vez definido un tipo, puede declarar una variable de ese tipo:

```
MiColorFavorito color;
```

La variable color puede contener uno de los valores definidos en el tipo enumerado MiColorFavorito o null, pero nada más. El tipo enumerado de Java es de tipo seguro, lo que significa que un intento de asignar un valor distinto de uno de los valores enumerados o null resultará en un error de compilación.

Se puede acceder a los valores enumerados utilizando la sintaxis

```
NombreTipoEnumerado.valueName
```

Por ejemplo, la siguiente sentencia asigna el valor enumerado AZUL a la variable color:

```
color = MiColorFavorito.AZUL;
```

Tenga en cuenta que debe utilizar el nombre del tipo enumerado como calificador para hacer referencia a un valor como AZUL.

Como con cualquier otro tipo, puedes declarar e inicializar una variable en una sola sentencia:

```
MiColorFavorito color = MiColorFavorito.AZUL;
```

Un tipo enumerado se trata como una clase especial. Por lo tanto, una variable de tipo enumerado es una variable de referencia. Un tipo enumerado es un subtipo de la clase Object y de la interfaz Comparable. Por lo tanto, un tipo enumerado hereda todos los métodos de la clase Object y el método de la interfaz Comparable. Además, puede utilizar los siguientes métodos en un objeto enumerado:

- `public String name();` Devuelve un nombre del valor para el objeto.
- `public int ordinal();` Devuelve el valor ordinal asociado al valor enumerado. El primer valor de un tipo enumerado tiene un valor ordinal 0, el segundo tiene un valor ordinal 1, el tercero 3, y así sucesivamente.

```
EnumeratedTypeDemo.java
1 public class EnumeratedTypeDemo {
2     static enum Dia {LUNES, MARTES, MIERCOLES, JUEVES,
3     VIERNES, SÁBADO, DOMINGO};
4
5     public static void main(String[] args) {
6         Dia dia1 = Dia.VIERNES;
7         Dia dia2 = Dia.JUEVES;
8
9         System.out.println("dia1 se llama " + dia1.name());
10        System.out.println("el nombre de dia2 es " + dia2.name());
11        System.out.println("el ordinal de dia1 es " + dia1.ordinal());
12        System.out.println("el ordinal de dia2 es " + dia2.ordinal());
13
14        System.out.println("dia1.equals(dia2) devuelve " +
15        dia1.equals(dia2));
16        System.out.println("dia1.toString() devuelve " +
17        dia1.toString());
18        System.out.println("dia1.compareTo(dia2) devuelve " +
19        dia1.compareTo(dia2));
20    }
21 }
```

dia1 se llama VIERNES  
dia2 se llama JUEVES  
el ordinal de dia1 es 4  
el ordinal de dia2 es 3  
dia1.equals(dia2) devuelve false  
dia1.toString() devuelve VIERNES  
dia1.compareTo(dia2) devuelve 1

En las líneas 2-3 se define un tipo enumerado Dia. Las variables dia1 y dia2 se declaran como de tipo Dia y se les asignan valores enumerados en las líneas 6-7. Como el valor de dia1 es VIERNES, su valor ordinal es 4 (línea 11). Como el valor de dia2 es JUEVES, su valor ordinal es 3 (línea 12).

Dado que un tipo enumerado es una subclase de la clase Object y de la interfaz Comparable, puedes invocar los métodos equals, toString, y compareTo desde un objeto enumerado (líneas 14-19).

day1.equals(day2) devuelve true si day1 y day2 tienen el mismo valor ordinal.

day1.compareTo(day2) devuelve la diferencia entre el valor ordinal de day1 y el de day2.