Enumeradores en Java

¿Qué es un enumerador (enum)?

Un enumerador (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 enumeradores respecto a una clase normal es que si tiene constructor debe ser privado para que no se puedan crear nuevos objetos.

Los enumeradores heredan de la clase Enum (java.lang.Enum) y por tanto tienen una serie de métodos heredados de esa clase padre.

Los enumeradores NO son Strings (aunque pueden serlo), sino que son objetos de una clase que solo pueden ser instanciados desde la clase que los implementa: no se puede crear un objeto de esa clase desde cualquier otro sitio que no sea dentro de esa clase y este proceso sucede al inicio de la ejecución del programa una sola vez. Es muy común 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.

Creación de un enumerador I

La declaración de un enumerador puede hacerse fuera de una clase, o dentro de una clase (class), pero NO dentro de un método.

Una enumeración se crea usando la palabra clave enum.
La primera línea dentro de un enum debe ser una lista de constantes y luego otras cosas como métodos, variables y constructores.

De acuerdo con las convenciones de nomenclatura de Java, se recomienda que nombremos las constantes con mayúsculas.

Creación de un enumerador II

```
public enum ColorAutomoviles {

    ROJO("Rojo"),
    AMARILLO("Amarillo"),
    AZUL("Azul"),
    BLANCO("Blanco"),
    GRIS("Gris Oscuro");

    private final String color;

    ColorAutomoviles(String color) { this.color = color;}

    public String getColor() { return color; }

}
```

Esta parte de la implementación es la que marca la diferencia "visual" con una clase

coma

Punto y coma (si la implementación no termina aquí). De lo contario no ponemos nada.

Esta propiedad es la que tendrá cada una de las instancias ROJO, AMARILLO, AZUL...con valores (por ejemplo) "Rojo", "Amarillo", "Azul"...
Como en cualquier clase podemos tener el número de atributos que queramos.

El contructor es igual al de cualquier clase

Cada atributo puede tener su Getter. No hay Setters dado que las instancias son final

Creación de un enumerador III

Pero ¿qué sucede en realidad en un enumerador cuando ejecutamos la aplicación?

```
class ColorAutomoviles {
    public static final ColorAutomoviles ROJO = new ColorAutomoviles();
    public static final ColorAutomoviles AMARILLO = new ColorAutomoviles();
    public static final ColorAutomoviles AZUL = new ColorAutomoviles();
    public static final ColorAutomoviles BLANCO = new ColorAutomoviles();
    public static final ColorAutomoviles GRIS = new ColorAutomoviles();
}
```

1 El enumerador se convierte en una clase

2 Crea tantos atributos (static y final) como los declarados en el enum. Estas variables son instancias de la clase misma y por lo tanto las crea llamando a su propio constructor!

3 Cada atributo es una instancia static y final del enum. Por lo tanto ES UN OBJETO, NO UN STRING!!!

Podríamos decir, a nivel coloquial, que un enum es una clase que genera un número "fijo" de instancias

Herencia en enumeradores

NO ES POSIBLE LA HERENCIA DE CLASES EN ENUMERADORES:

1) NO SE PUEDE EXTENDER DE ÉL: UN ENUMERADOR POR DEFECTO ES FINAL
2) NO PUEDE EXTENDER DE OTRA CLASE YA QUE POR DEFECTO HEREDA DE LA CLASE ENUM Y EN JAVA NO EXISTE LA HERENCIA MÚLTIPLE

LA HERENCIA DE INTERFACES ESTÁ PERMITIDA EN ENUMERADORES: UN ENUMERADOR PUEDE IMPLEMENTAR UNA INTERFACE