

Enum

Qué son, para qué sirven y cómo funcionan los enumeradores.

Definición

Los **Enum** o **enumerados** son una estructura de datos que permite definir un conjunto de constantes con **nombre**. Estas constantes representan valores que son fijos y predefinidos (utilizando el operador *final*), lo que significa que no pueden ser modificados durante la ejecución del programa. Son útiles porque nos permiten definir un conjunto de opciones o estados posibles para una variable o un objeto, de una manera clara y concisa. Por ejemplo, podemos usar un Enum para representar los días de la semana, los meses del año, las opciones de un menú, los estados de un pedido, y así sucesivamente.

Constantes definidas de esta manera hacen el código más legible, permiten verificación en tiempo de compilación, documentan por adelantado la lista de valores aceptados y evitan el comportamiento inesperado si es que se reciben valores no válidos. Un tipo enumerado restringe los posibles valores que puede tomar una variable. Esto ayuda a reducir los errores en el código y permite algunos usos especiales interesantes.

Sintaxis y características

Los Enum en Java funcionan de manera similar a las clases, ya que también pueden tener métodos, constructores (solo privados) y variables miembro (atributos). Sin embargo, a diferencia de las clases, los Enum tienen un número limitado de instancias, que son todas las constantes que definimos en su declaración. Dicho de otra manera, **cada constante es un objeto (instancia) del enumerado**. Las constantes se escriben en mayúscula y separadas por guiones bajos. En la siguiente imagen tenemos un grupo de constantes que representan ciudades argentinas. Cada constante es una instancia del enumerador *Ciudades*.

```
public enum Ciudades {  
    MAR_DEL_PLATA, BUENOS_AIRES, PINAMAR, NECOCHEA, MAR_CHIQUITA,  
    MIRAMAR  
}
```

Una vez definidas las constantes, las podemos utilizar de la siguiente manera:

Ciudades miCiudad = **Ciudades** .MAR_DEL_PLATA;

A la variable *miCiudad* de tipo **Ciudades** le estamos asignando el valor “MAR DEL PLATA” (no es un String). Recordemos que no puede ser modificado en tiempo de ejecución.

Si le queremos agregar atributos a cada constante, debemos hacerlo agregando paréntesis luego de la constante e indicando el dato que va a alojar. En el siguiente ejemplo se le agrega el nombre de la ciudad a cada constante, cambiando un poco la forma en que estaba definida anteriormente:

```
public enum Ciudades {  
  
    MDP("Mar del Plata"),  
    BSAS("Buenos Aires"),  
    PIN("Pinamar"),  
    NEC("Necochea"),  
    MARCHI("Mar Chiquita"),  
    MIR("Miramar");  
  
...}
```

Ahora cada constante va a tener un atributo de tipo String que indique el nombre de la ciudad.

Como cada constante es una instancia del enumerador *Ciudades* se deben tener dos cosas importantes definidas, el o los **atributos** y el **constructor** con sus respectivos modificadores de accesos y parámetros:

- **Los atributos deben ser privados y finales**, ya que si es una constante se supone que su valor no cambia. Debe haber un método getter para poder acceder a los atributos de cada constante.
- **El constructor del enumerador tiene que ser privado**, ya que, desde un inicio, cuenta con todas las instancias posibles (cada constante). Si no definimos un constructor, Java crea uno por defecto teniendo en cuenta a cada atributo.
- **El numerador puede tener métodos, ya sean estáticos o no**. Si no son estáticos, van a trabajar con los atributos de la instancia. Por ejemplo, si trabajamos con la constante MDP, trabajamos con el atributo “Mar del plata” y no con “Necochea”.
- **Un tipo enumerado puede ser declarado dentro o fuera de una clase, pero no dentro de un método**. Por tanto no podemos declarar un enum dentro de un método main, constructor u otros.
- **Se dispone automáticamente de métodos especiales como por ejemplo el método values()**, que el compilador agrega automáticamente cuando se crea un enum. Este método devuelve un array conteniendo todos los valores del enumerado en el orden en que son declarados.

Si agregamos un atributo más al enumerador nos quedaría de la siguiente manera:

```
public enum Ciudades {  
  
    MDP("Mar del Plata", "593.337"),  
    BSAS("Buenos Aires", "3.075.646"),  
    PIN("Pinamar", "25.397"),  
    NEC("Necochea", "92.933"),  
    MARCHI("Mar Chiquita", "487"),  
    MIR("Miramar", "29.433 ");  
  
    private final String ciudad;  
    private final String habitantes;  
  
    Ciudades (String ciudad, String habitantes) {  
  
        this.ciudad = ciudad;  
        this.habitantes = habitantes;  
    }  
  
    public String getCiudad() {  
        return ciudad;  
    }  
  
    public String getHabitantes() {  
        return habitantes;  
    }  
  
    @Override  
    public String toString() {  
        return ciudad + ", habitantes: " + habitantes;  
    }  
}
```