

# CURSO DE PROGRAMACIÓN CON JAVA

# ENUMERACIONES EN JAVA



Por el experto: Ing. Ubaldo Acosta



**CURSO DE PROGRAMACIÓN CON JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de enumeraciones en Java.

¿Estás listo? Ok, ¡Vamos!

## ENUMERACIONES EN JAVA

### Definición de una Enumeración en Java:

```
public enum Dias {  
  
    //Son valores constantes, por eso van en mayusculas  
    LUNES,  
    MARTES,  
    MIERCOLES,  
    JUEVES,  
    VIERNES,  
    SABADO,  
    DOMINGO  
}
```

### CURSO DE PROGRAMACIÓN CON JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Los tipos enumerados son un tipo especial de clase, el cual nos permite asociar una serie de valores de tipo final, es decir que son constantes. Por ejemplo podemos observar en la lámina las constantes de los días de la semana.

Vemos que es como crear una clase de Java, solo que en lugar de la palabra `class`, utilizamos la palabra reservada `enum`.

Debido a que los elementos de una enumeración son constantes, se escriben con mayúsculas, como hemos comentado que es una buena práctica cuando usamos constantes en Java.

Así que podemos utilizar una clase de tipo enumeración cada vez que necesitemos declarar una serie de valores constantes y que conozcamos previamente, como los días de la semana, los puntos cardinales, el conjunto de planetas, continentes, y en si cualquier conjunto de datos que son constantes.

# USO DE ENUMERACIONES EN JAVA

## Uso de una Enumeración en Java:

```
public class EjemploEnumeraciones {  
  
    public static void main(String[] args) {  
        //Valores de la enumeracion  
        System.out.println("Valor 1:" + Dias.LUNES);  
  
        //Hacemos un test del dia utilizado  
        indicarDia(Dias.LUNES);  
    }  
  
    public static void indicarDia(Dias dias) {  
        switch (dias) {  
            //Podemos usar algun valor constante de la enumeracion directamente  
            case LUNES:  
                System.out.println("Primer dia de la semana");  
                break;  
        }  
    }  
}
```

Una vez que hemos definido nuestro tipo enumeración podemos utilizarlo como se ve en la lámina.

Lo que hacemos es indicar el nombre de la enumeración, en este caso Dias, y posteriormente utilizamos alguno de los valores constantes de la enumeración, por ejemplo: Dias.LUNES, con eso estaremos utilizando alguno de los valores de enumeración según necesitemos.

Sin embargo no es su único uso, podemos utilizar una enumeración para comprobar algunos de los casos de una estructura switch, la cual sabemos que seleccionará alguno de los casos según coincida con la entrada de la sentencia switch.

Para probar alguno de los casos de la estructura switch, debemos proporcionar ya un valor de la enumeración, como puede ser Dias.LUNES o Dias.MARTES, etc.

Y dentro del switch en cada uno de los casos podemos especificar los valores ya sin especificar el nombre de la enumeración, sino simplemente indicando la constante de la enumeración, por ejemplo case LUNES, o case MARTES, etc.

Este es sólo un ejemplo de cómo podemos utilizar las enumeraciones.

# ATRIBUTOS Y METODOS EN UNA ENUMERACION

## Atributos y métodos en una enumeración:

```
public enum Continentes {  
    AFRICA(53),  
    EUROPA(46),  
    ASIA(44),  
    AMERICA(34),  
    OCEANIA(14);  
  
    //Atributo de cada elemento de una enumeracion  
    private final int paises;  
  
    //Constructor de cada elemento de la enumeracion  
    Continentes(int paises) {  
        this.paises = paises;  
    }  
  
    public int getPaises() {  
        return paises;  
    }  
}
```

Sin embargo el tipo enumeración es más complejo que eso. Podemos además de definir simples valores constantes, valores a cada una de las constantes.

Para ello debemos crear un constructor para inicializar cada uno de los valores asociado a cada constante.

Por ejemplo en el código podemos observar que estamos creando una enumeración de los continentes existentes, sin embargo nos interesaría indicar el número de países existentes por cada continente.

Para ello cuando definimos cada constante de la enumeración, podemos proporcionar los valores que deseemos separados por comas, en este caso es sólo un valor entero el que estamos proporcionado, el cual significa el número de países por cada continente que declaramos.

Sin embargo para poder crear este tipo de enumeraciones más complejas, necesitamos crear un constructor para poder crear cada uno de los elementos de la enumeración, y además los atributos que se asociarán con los valores proporcionados por cada constante de la enumeración.

Por ello vemos en el código, que no basta con crear las constantes de la enumeración, sino que debemos además declarar el atributo países de tipo entero, y posteriormente declaramos un constructor, el cual se llamará de manera automática por cada uno de los elementos de la enumeración.

Y finalmente, debido a que cada uno de los elementos de la enumeración contiene definido el numero de países, es posible declarar un método getPaises para recuperar precisamente el atributo países declarado en el tipo enumeración.

A continuación vamos a ver cómo utilizar una enumeración de este tipo.

## USO DE LA ENUMERACION

### Uso de la enumeración con atributos y métodos:

```
public static void main(String[] args) {  
  
    //Utilizamos la enumeracion de paises  
    System.out.println("Continente no. 4: " + Continentes.AMERICA);  
    System.out.println("Paises en America: " + Continentes.AMERICA.getPaises());  
  
    //Hacemos el test del numero de paises por continente  
    System.out.println("");  
    indicarPaises(Continentes.AFRICA);  
}  
  
public static void indicarPaises(Continentes continentes) {  
  
    switch (continentes) {  
        //Podemos usar algun valor constante de la enumeracion directamente  
        case AFRICA:  
            System.out.println("No. Paises en:" + continentes  
            + ": " + continentes.getPaises());  
    }  
}
```

Como podemos observar, el uso básico es de las constantes de una enumeración es el mismo que hemos estudiado, por ejemplo, para acceder a alguna de las constantes especificamos el nombre de la enumeración seguido del nombre de la constante, por ejemplo: Continentes.AMERICA, y si deseamos acceder a los atributos de cada constante definida, podemos llamar el método para recuperar el atributo de cada constante, por ejemplo: Contienentes.AMERICA.getPaises()

Con esto podemos observar que el uso del tipo enumeraciones puede ser más completo que simplemente asociar algunas constantes, sino que también podemos definir atributos y por tanto agregar métodos para recuperar los atributos para cada una de las constantes definidas de una enumeración.

## VALORES DE UNA ENUMERACION

### Valores de una enumeración:

```
public class EjemploEnumeraciones {  
  
    public static void main(String[] args) {  
  
        //Imprimimos los continentes  
        System.out.println("");  
        imprimirContinentes();  
    }  
  
    public static void imprimirContinentes() {  
        //Utilizamos un forEach  
        for(Continentes c: Continentes.values()) {  
            System.out.println("Continente:" + c  
                + " contiene " + c.getPaises() + " paises.");  
        }  
    }  
}
```

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

El método values se agrega por default cada que creamos una enumeración. El objetivo de este método es que nos regrese una lista de cada una de las constantes definidas en una enumeración.

En el código podemos ver en el método imprimirContinentes, el uso de este método, apoyados de un ciclo forEach, lo que hacemos es definir una variable de tipo enumeración que estas utilizando, en este caso Continentes c, y posteriormente indicamos los valores de la enumeración llamando precisamente al método Continentes.values().

Una vez que estamos iterando cada uno de los elementos, tenemos la variable c para recuperar el atributo de la constante que estamos iterando, por ello podemos pedir el método c.getPaises para recuperar el número de países del continente seleccionado y si queremos imprimir el continente que estamos iterando simplemente usamos la variable c definida en el ciclo forEach.

A continuación veamos un ejemplo del uso de enumeraciones.

## EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio de Enumeraciones en Java.

**CURSO DE PROGRAMACIÓN CON JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# CURSO ONLINE

# PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



## CURSO DE PROGRAMACIÓN CON JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>✓ Lógica de Programación</li><li>✓ Fundamentos de Java</li><li>✓ Programación con Java</li><li>✓ Java con JDBC</li><li>✓ HTML, CSS y JavaScript</li><li>✓ Servlets y JSP's</li><li>✓ Struts Framework</li></ul> | <ul style="list-style-type: none"><li>✓ Hibernate Framework</li><li>✓ Spring Framework</li><li>✓ JavaServer Faces</li><li>✓ Java EE (EJB, JPA y Web Services)</li><li>✓ JBoss Administration</li><li>✓ Android con Java</li><li>✓ HTML5 y CSS3</li></ul> |
|---|--|

### Datos de Contacto:

Sitio Web: [www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Email: [informes@globalmentoring.com.mx](mailto:informes@globalmentoring.com.mx)

