

## CURSO DE PROGRAMACIÓN CON JAVA

# SOBREESCRITURA DE METODOS EN JAVA



Ing. Ubaldo Acosta

Por el experto: Ing. Ubaldo Acosta



## CURSO DE PROGRAMACIÓN CON JAVA

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

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

Vamos a estudiar el tema de sobreescritura de métodos en Java.

¿Estás listo? ¡Vamos!

## SOBREESCRITURA EN LA VIDA REAL



Madre



Hija

**CURSO DE PROGRAMACIÓN CON JAVA**

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

Vamos a estudiar ahora el concepto de sobreescritura de métodos en Java. Veamos primero el concepto como sucede en el mundo real y posteriormente explicaremos cómo aplicar este concepto en código Java.

La sobreescritura de métodos en automático tiene que ver con el concepto de herencia, y se trata básicamente de la modificación de un método desde una clase hija a un método heredado por la clase padre.

Por ejemplo, en la vida real, podemos observar que el talento, por ejemplo cantar, se hereda de madre a hija, sin embargo, aunque la hija puede cantar similar a la madre debido a que ha heredado varias características de la madre, no lo hará exactamente igual, por lo tanto, adecua el canto a la forma de la hija y su capacidad para explotar este talento.

De igual manera la sobreescritura de métodos tiene que ver con la modificación de un método por parte de la clase hija, de algún método que ha sido heredado por la clase padre, de allí el nombre de sobreescritura, ya que el hijo vuelve a escribir nuevamente el comportamiento del método heredado para adaptarlo a sus propias necesidades.

# SOBREESCRITURA DE MÉTODOS EN JAVA



```

1 public class Empleado {
2     protected String nombre;
3     protected String puesto;
4     protected int nivel;
5
6     public String obtenerDetalles(){
7         return "Nombre: " + nombre + "\n" +
8             "Puesto: " + puesto + "\n" +
9             "Nivel: " + nivel;
10    }
11 }
  
```

```

1 public class Gerente extends Empleado{
2     private String departamento;
3
4     public String obtenerDetalles(){
5         return "Nombre : " + nombre + "\n" +
6             "Puesto : " + puesto + "\n" +
7             "Nivel : " + nivel + "\n" +
8             "Departamento: " + departamento ;
9     }
10 }
11
  
```

## CURSO DE PROGRAMACIÓN CON JAVA

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

Como hemos comentado, la sobreescritura de métodos aplica si existe una clase padre y una clase hija. Como observamos en la figura, tenemos dos clases, una clase Padre llamada Empleado, y una clase hija llamada Gerente. Y vemos un ejemplo de código de ambas clases.

Por un lado la clase padre, declara 3 atributos de tipo `protected`, esto quiere decir, que son heredados y accesibles directamente por cualquier clase hija. Y además la clase Empleado define un método llamado `obtenerDetalles()`. Este método básicamente regresa una cadena concatenando cada uno de los atributos de la clase, esto es muy similar a lo que hace el método `toString()` el cual estudiaremos a detalle más adelante.

Por otro lado tenemos a la clase hija llamada Gerente, y como vemos hace `extends` de la clase Empleado, por lo tanto hereda los atributos de tipo `protected` y también el método público llamado `obtenerDetalles()`. Sin embargo, la clase Gerente define un atributo que no está definido en la clase padre, llamado `departamento`, y por lo tanto este atributo no se desplegará si utilizamos directamente el método `obtenerDetalles()` de la clase padre. Esta es a grandes rasgos la razón por la cual en la programación orientada a objetos es necesario la sobreescritura de métodos, y en este caso en particular la clase Gerente redefine o sobreescribe el método `obtenerDetalles()`, y con ello lo que hace es agregar el comportamiento para que el método incluya el atributo `departamento`, y con ello se complete el funcionamiento deseado desde la clase Gerente.

Así que como hemos visto, una subclase puede modificar el comportamiento heredado de una clase padre.

La firma del método sobreescrito por una subclase debe ser igual que el padre en: **nombre, tipo de retorno y lista de argumentos**, esto es una regla para que se cumpla y respete el concepto de sobreescritura.

En la firma del método lo único que puede variar es el modificador de acceso, pero un método sobreescrito no puede ser menos accesible que el método que sobreescribe, por ejemplo, si el método padre es `public`, no puede cambiarse a `protected` en el hijo, y mucho menos a `private`.

# EJERCICIOS CURSO PROGRAMACIÓN CON JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Sobreescritura de Métodos 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



Experiencia y Conocimiento para tu vida

**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)