

CURSO DE PROGRAMACIÓN CON JAVA

EJERCICIO

SOBREESCRITURA DE MÉTODOS EN JAVA



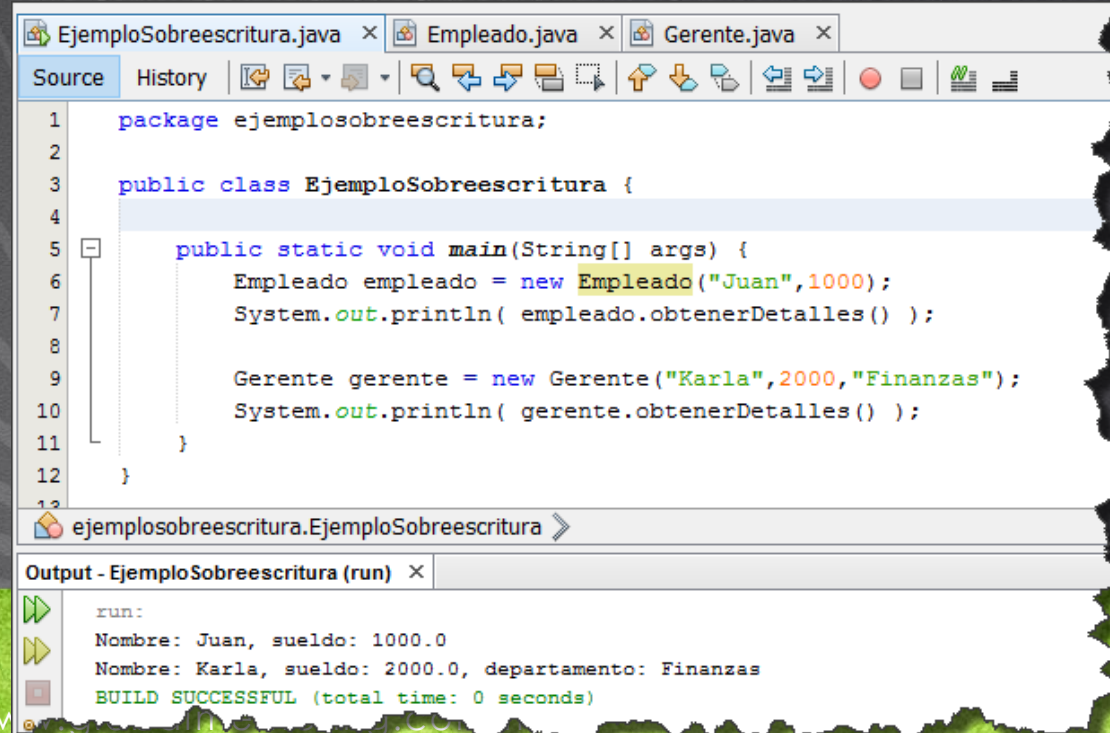
Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear un programa para poner en práctica el concepto de sobrescritura de métodos en Java. Al finalizar deberemos observar lo siguiente:



The screenshot shows an IDE with three tabs: EjemploSobreescritura.java, Empleado.java, and Gerente.java. The main editor displays the source code for EjemploSobreescritura.java. The code defines a package, a public class, and a main method that creates instances of Empleado and Gerente, printing their details. The output window at the bottom shows the execution results, confirming that the program ran successfully and printed the details for Juan and Karla.

```
1 package ejemplosobreescritura;
2
3 public class EjemploSobreescritura {
4
5     public static void main(String[] args) {
6         Empleado empleado = new Empleado("Juan",1000);
7         System.out.println( empleado.obtenerDetalles() );
8
9         Gerente gerente = new Gerente("Karla",2000,"Finanzas");
10        System.out.println( gerente.obtenerDetalles() );
11    }
12 }
13
```

ejemplosobreescritura.EjemploSobreescritura

Output - EjemploSobreescritura (run)

```
run:
Nombre: Juan, sueldo: 1000.0
Nombre: Karla, sueldo: 2000.0, departamento: Finanzas
BUILD SUCCESSFUL (total time: 0 seconds)
```

CURSO

WV

PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto:

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: EjemploSobreescritura

Project Location: C:\Cursos\ProgramacionJava\Leccion10 Browse...

Project Folder: rogramacionJava\Leccion10\EjemploSobreescritura

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class ejemplosobreescritura.EjemploSobreescritura

< Back Next > **Finish** Cancel Help

PASO 2. CREACIÓN DE LA CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 3. CREACIÓN DE LA CLASE

Vamos a crear una clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Empleado.java:

```
package ejemplosobreescritura;

public class Empleado {
    protected String nombre;
    protected double sueldo;

    protected Empleado(String nombre, double sueldo){
        this.nombre = nombre;
        this.sueldo = sueldo;
    }

    public String obtenerDetalles(){
        return "Nombre: " + nombre +
            ", sueldo: " + sueldo;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public double getSueldo() {
        return sueldo;
    }

    public void setSueldo(double sueldo) {
        this.sueldo = sueldo;
    }
}
```

PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Gerente.java:

```
package ejemplosobreescritura;

public class Gerente extends Empleado{

    private String departamento;

    public Gerente(String nombre, double sueldo, String departamento) {
        super(nombre, sueldo);
        this.departamento = departamento;
    }

    //Sobreescribimos el metodo padre heredado
    public String obtenerDetalles(){
        //Observar como accedemos directamente al atributo heredado
        //debido a que se declaro como protected en la clase padre
        //y por tanto la clase hija lo hereda y accede directamente
        return "Nombre: " + nombre +
            ", sueldo: " + sueldo +
            ", departamento: " + departamento;
    }

    public String getDepartamento() {
        return departamento;
    }

    public void setDepartamento(String departamento) {
        this.departamento = departamento;
    }

}
```

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo EjemploSobreescritura.java:

```
package ejemplosobreescritura;

public class EjemploSobreescritura {

    public static void main(String[] args) {
        Empleado empleado = new Empleado("Juan",1000);
        System.out.println( empleado.obtenerDetalles() );

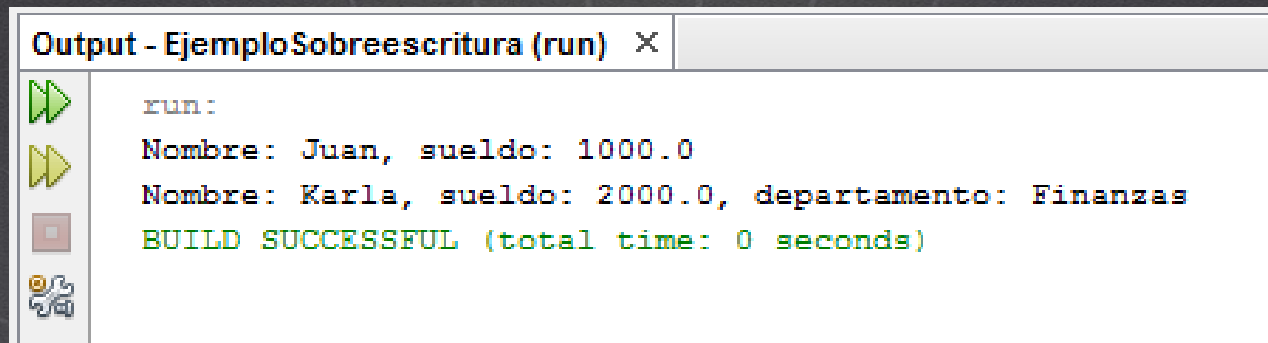
        Gerente gerente = new Gerente("Karla",2000,"Finanzas");
        System.out.println( gerente.obtenerDetalles() );
    }
}
```

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

PASO 7. EJECUTAMOS EL PROYECTO

El resultado es como sigue:



The screenshot shows an IDE output window titled "Output - EjemploSobreescritura (run)". The window contains the following text:

```
run:  
Nombre: Juan, sueldo: 1000.0  
Nombre: Karla, sueldo: 2000.0, departamento: Finanzas  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The output window has a standard toolbar on the left with icons for running, stepping through code, and other debugging actions.

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Probar con distintos valores y verificar el resultado.



Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica el concepto de sobreescritura de métodos en Java.
- Hemos visto cómo la sobreescritura es un concepto sencillo pero poderoso en la programación orientada a objetos, por lo que su aplicación y claro entendimiento nos permitirá avanzar más rápidamente en el aprendizaje del lenguaje Java.



CURSO DE PROGRAMACIÓN CON JAVA

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