

CURSO DE PROGRAMACIÓN CON JAVA

EJERCICIO

EJERCICIO CLASE OBJECT Y METODOS TOSTRING, EQUALS Y HASHCODE



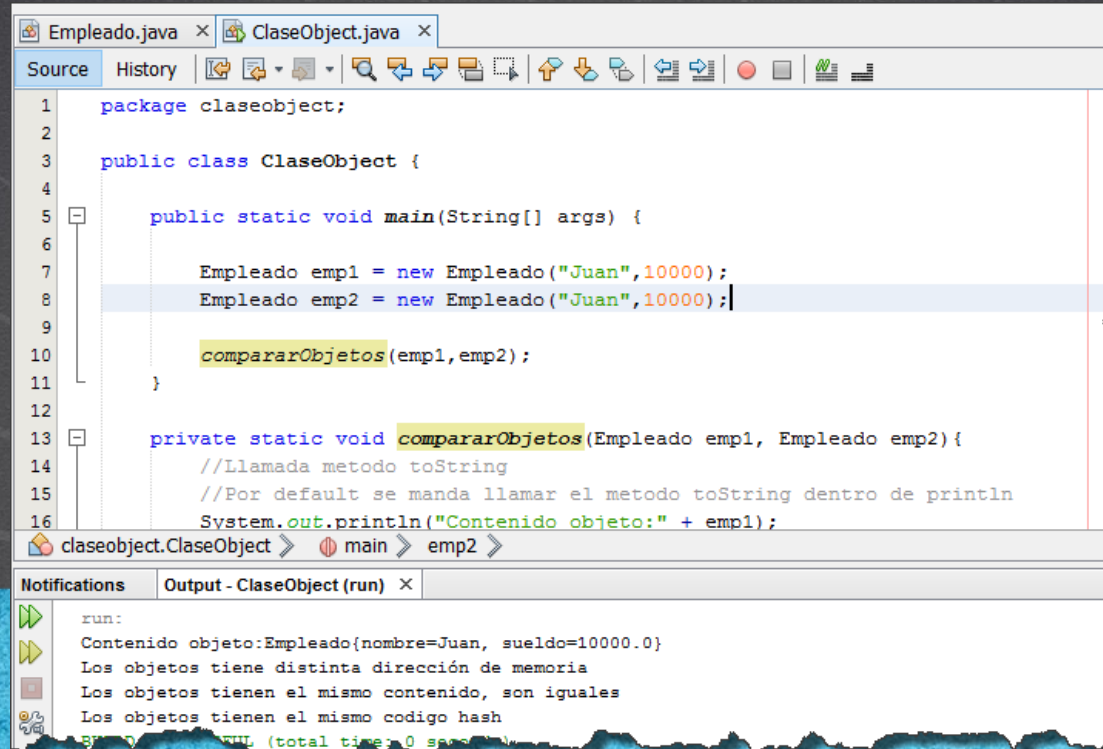
Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear el ejercicio de la Clase Object. Al finalizar deberemos observar lo siguiente:



```
1 package claseobject;
2
3 public class ClaseObject {
4
5     public static void main(String[] args) {
6
7         Empleado emp1 = new Empleado("Juan",10000);
8         Empleado emp2 = new Empleado("Juan",10000);
9
10        compararObjetos(emp1,emp2);
11    }
12
13    private static void compararObjetos(Empleado emp1, Empleado emp2){
14        //Llamada metodo toString
15        //Por default se manda llamar el metodo toString dentro de println
16        System.out.println("Contenido objeto:" + emp1);
17    }
18 }
```

claseobject.ClaseObject > main > emp2 >

Notifications Output - ClaseObject (run) ×

```
run:
Contenido objeto:Empleado{nombre=Juan, sueldo=10000.0}
Los objetos tienen distinta dirección de memoria
Los objetos tienen el mismo contenido, son iguales
Los objetos tienen el mismo código hash
```

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:

Project Location: Browse...

Project Folder:

☐ 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

< Back Next > **Finish** Cancel Help

PASO 2. CREACIÓN DE UNA 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. MODIFICAMOS EL CÓDIGO

Archivo Empleado.java:

```
package claseobject;

public class Empleado {

    private String nombre;

    private double sueldo;

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

```
@Override
public String toString() {
    return "Empleado{" + "nombre=" + nombre + ", sueldo=" + sueldo + '}';
}

@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (obj instanceof Empleado) {
        Empleado emp = (Empleado) obj;
        if (nombre.equals(emp.nombre) &&
Double.valueOf(sueldo).equals(emp.sueldo)) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

@Override
public int hashCode() {
    int hash = 7;
    hash = 31 * hash + this.nombre.hashCode();
    hash = 31 * hash + Double.valueOf(this.sueldo).hashCode();
    return hash;
}
}
```

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo ClaseObject.java:

```
package claseobject;

public class ClaseObject {

    public static void main(String[] args) {

        Empleado emp1 = new Empleado("Juan",10000);
        Empleado emp2 = new Empleado("Juan",10000);

        compararObjetos(emp1,emp2);

    }
```

```
private static void compararObjetos(Empleado emp1, Empleado emp2){
    //Llamada metodo toString
    //Por default se manda llamar el metodo toString dentro de println
    System.out.println("Contenido objeto:" + emp1);

    //Revision por referencia
    if( emp1 == emp2)
        System.out.println("Los objetos tiene misma dirección de memoria");
    else
        System.out.println("Los objetos tiene distinta dirección de memoria");

    //Revision por el metodo equals
    if(emp1.equals(emp2))
        System.out.println("Los objetos tienen el mismo contenido, son iguales");
    else
        System.out.println("Los objetos NO tienen el mismo contenido, NO son iguales");

    //Revisamos el metodo hashCode
    if(emp1.hashCode() == emp2.hashCode())
        System.out.println("Los objetos tienen el mismo codigo hash");
    else
        System.out.println("Los objetos NO tienen el mismo codigo hash");

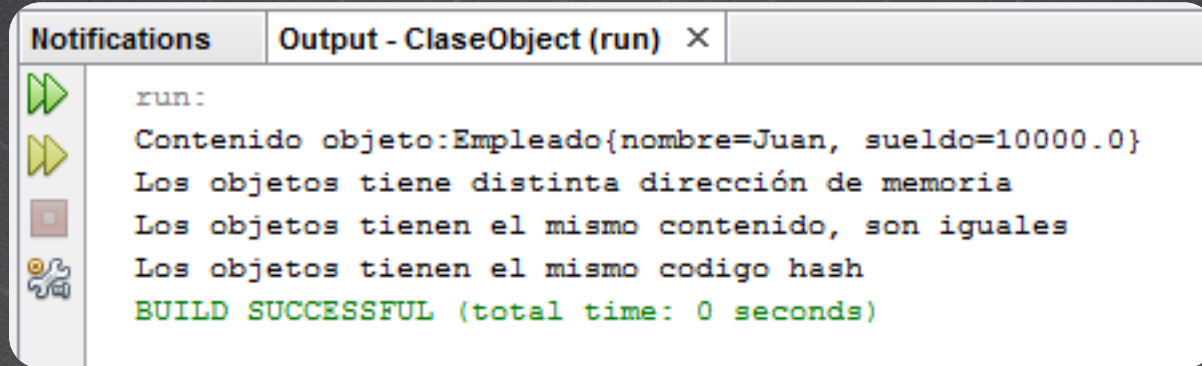
}
}
```

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

PASO 5. EJECUTAMOS EL PROYECTO

El resultado es como sigue:



The screenshot shows an IDE's Output window with the title "Output - ClaseObject (run)". The window contains the following text:

```
run:
Contenido objeto:Empleado{nombre=Juan, sueldo=10000.0}
Los objetos tiene distinta dirección de memoria
Los objetos tienen el mismo contenido, son iguales
Los objetos tienen el mismo código hash
BUILD SUCCESSFUL (total time: 0 seconds)
```

On the left side of the window, there are four icons: a green play button, a yellow play button, a red square, and a blue icon with a magnifying glass.

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 manejo de la clase Object, así como la sobreescritura de algunos de los métodos más importantes, como son los métodos toString, equals y hashCode.
- Para más información consultar:
- <https://docs.oracle.com/javase/tutorial/java/landl/objectclass.html>

CURSO ONLINE

PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx