

CURSO DE FUNDAMENTOS DE JAVA

EJERCICIO

HERENCIA EN JAVA



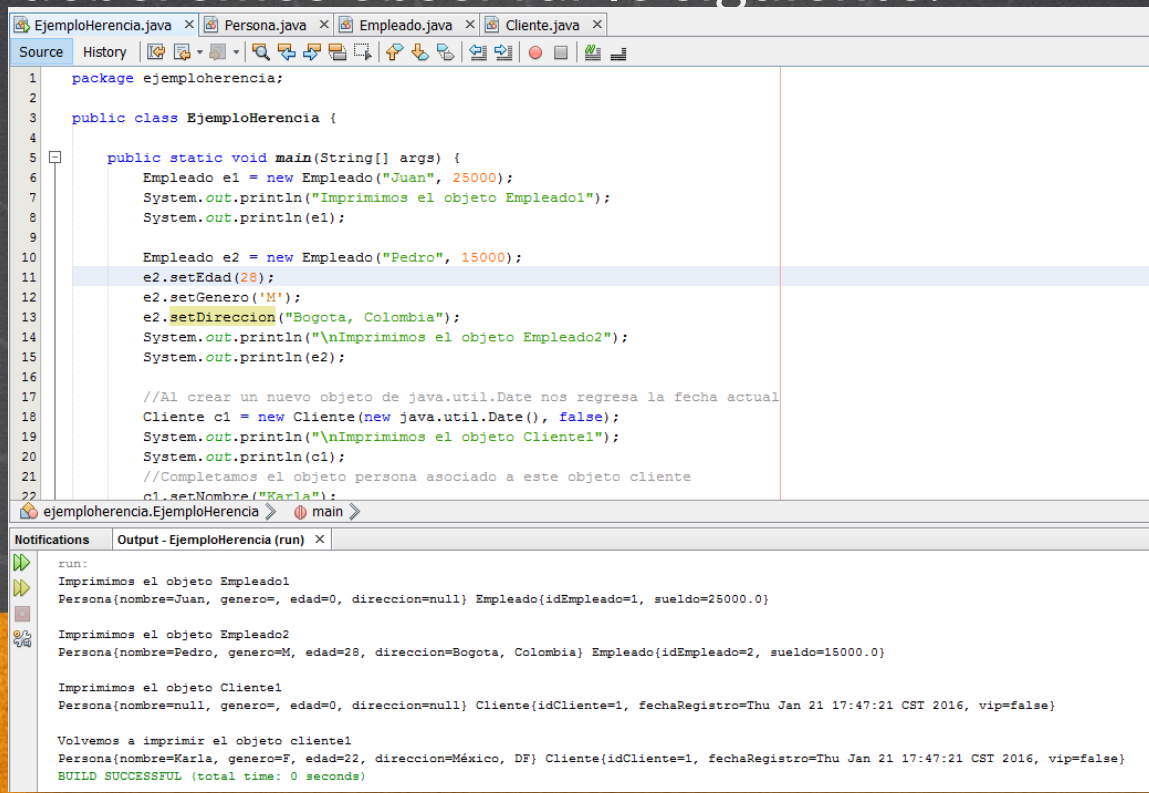
Experiencia y Conocimiento para tu vida

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear un ejercicio para aplicar el concepto de Herencia en Java.
Al finalizar deberemos observar lo siguiente:



The screenshot shows an IDE with four tabs: EjemploHerencia.java, Persona.java, Empleado.java, and Cliente.java. The 'Source' view displays the code for EjemploHerencia.java. The code defines a package 'ejemploherencia' and a public class 'EjemploHerencia' with a main method. The main method creates three objects: 'Empleado e1' (Juan, salary 25000), 'Empleado e2' (Pedro, salary 15000, age 28, gender 'M', direction 'Bogota, Colombia'), and 'Cliente c1' (new java.util.Date(), false). It prints details for each object. A comment indicates that creating a new Date object returns the current date. The 'Output - EjemploHerencia (run)' view shows the execution results, including the object details and a successful build message.

```
1 package ejemploherencia;
2
3 public class EjemploHerencia {
4
5     public static void main(String[] args) {
6         Empleado e1 = new Empleado("Juan", 25000);
7         System.out.println("Imprimimos el objeto Empleado1");
8         System.out.println(e1);
9
10        Empleado e2 = new Empleado("Pedro", 15000);
11        e2.setEdad(28);
12        e2.setGenero('M');
13        e2.setDireccion("Bogota, Colombia");
14        System.out.println("\nImprimimos el objeto Empleado2");
15        System.out.println(e2);
16
17        //Al crear un nuevo objeto de java.util.Date nos regresa la fecha actual
18        Cliente c1 = new Cliente(new java.util.Date(), false);
19        System.out.println("\nImprimimos el objeto Cliente1");
20        System.out.println(c1);
21        //Completamos el objeto persona asociado a este objeto cliente
22        c1.setNombre("Karla");
23    }
24}
```

ejemploherencia.EjemploHerencia > main >

run:

```
Imprimimos el objeto Empleado1
Persona{nombre=Juan, genero=, edad=0, direccion=null} Empleado{idEmpleado=1, sueldo=25000.0}

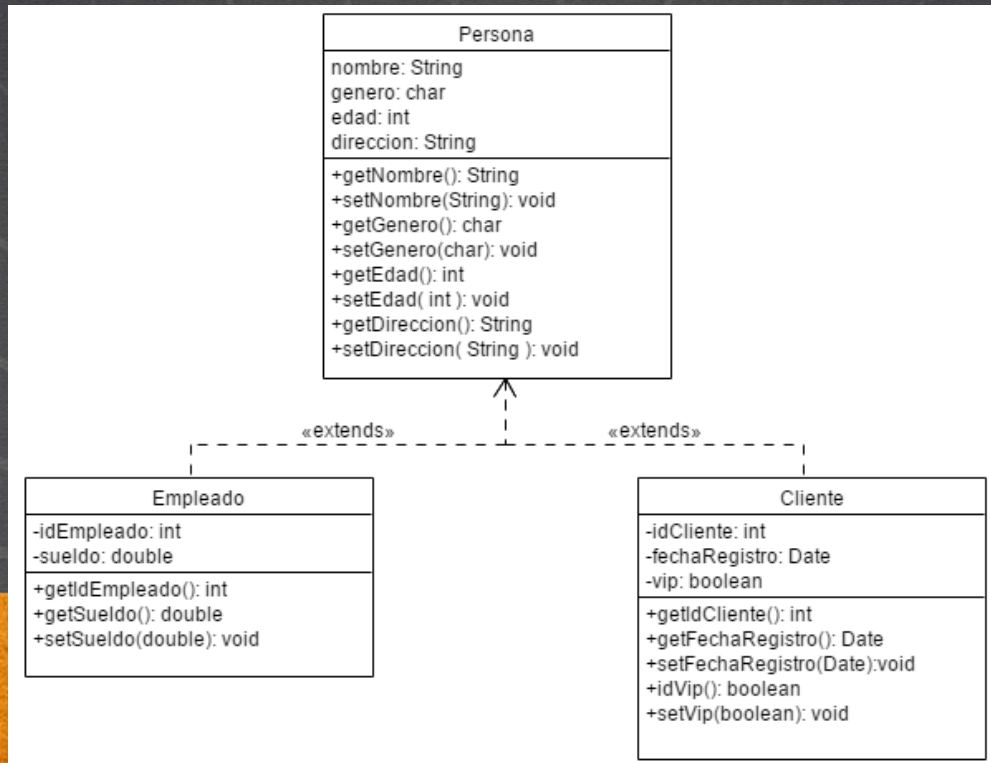
Imprimimos el objeto Empleado2
Persona{nombre=Pedro, genero=M, edad=28, direccion=Bogota, Colombia} Empleado{idEmpleado=2, sueldo=15000.0}

Imprimimos el objeto Cliente1
Persona{nombre=null, genero=, edad=0, direccion=null} Cliente{idCliente=1, fechaRegistro=Thu Jan 21 17:47:21 CST 2016, vip=false}

Volvemos a imprimir el objeto cliente1
Persona{nombre=Karla, genero=F, edad=22, direccion=México, DF} Cliente{idCliente=1, fechaRegistro=Thu Jan 21 17:47:21 CST 2016, vip=false}
BUILD SUCCESSFUL (total time: 0 seconds)
```

DIAGRAMA DE CLASES

El siguiente es un diagrama de Clases del ejercicio, creado con la herramienta <http://www.umlet.com/umletino/umletino.html>:



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. CREAMOS LA CLASE PERSONA

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

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

PASO 3. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

```
package ejemploherencia;

public class Persona { //Definición de la clase Padre
    private String nombre;
    private char genero;
    private int edad;
    private String direccion;

    //Constructor vacio
    public Persona(){}

    //Constructor 1 argumento
    public Persona(String nombre){
        this.nombre = nombre;
    }

    //Constructor completo
    public Persona(String nombre, char genero, int edad, String direccion){
        this.nombre = nombre;
        this.genero = genero; //M-Masculino, F-Femenino
        this.edad = edad;
        this.direccion = direccion;
    }

    public String getNombre() {
        return nombre;
    }

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

    public char getGenero() {
        return genero;
    }

    public void setGenero(char genero) {
        this.genero = genero;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

    @Override
    public String toString() {
        return "Persona{" + "nombre=" + nombre + ", genero=" + genero + ", edad=" + edad + ", direccion=" + direccion + '}';
    }
}
```

PASO 4. CREAMOS LA CLASE EMPLEADO

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

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Empleado.java:

```
package ejemploherencia;

public class Empleado extends Persona {

    private int idEmpleado;
    private double sueldo;
    private static int contadorEmpleados;

    public Empleado(String nombre, double sueldo) {
        super(nombre); //Super debe ser la primera linea
        this.idEmpleado = ++contadorEmpleados;
        this.sueldo = sueldo;
    }

    public int getIdEmpleado() {
        return idEmpleado;
    }
}
```

```
public double getSueldo() {
    return sueldo;
}

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

@Override
public String toString() {
    //Primero mandamos a llamar el método toString de la clase
    Persona
    //para que podamos observar los valores de la clase Padre,
    //y despues imprimimos los valores de la clase hija
    return super.toString() + " Empleado{" + "idEmpleado=" +
    idEmpleado + ", sueldo=" + sueldo + '}';
}
}
```


PASO 6. CREAMOS LA CLASE CLIENTE

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo Cliente.java:

```
package ejemploherencia;

import java.util.Date;

public class Cliente extends Persona{
    private int idCliente;
    private java.util.Date fechaRegistro;
    private boolean vip;
    private static int contadorClientes;

    public Cliente(Date fechaRegistro, boolean vip) {
        this.idCliente = ++contadorClientes;
        this.fechaRegistro = fechaRegistro;
        this.vip = vip;
    }

    public int getIdCliente() {
        return idCliente;
    }
}
```

```
public Date getFechaRegistro() {
    return fechaRegistro;
}

public boolean isVip() {
    return vip;
}

public void setVip(boolean vip) {
    this.vip = vip;
}

@Override
public String toString() {
    //Primero mandamos a llamar el método toString de la clase
    Persona
    //para que podamos observar los valores de la clase Padre,
    //y despues imprimimos los valores de la clase hija
    return super.toString() + " Cliente{" + "idCliente=" +
    idCliente + ", fechaRegistro=" + fechaRegistro + ", vip=" + vip +
    '}';
}
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo EjemploHerencia.java:

```
package ejemploherencia;

public class EjemploHerencia {

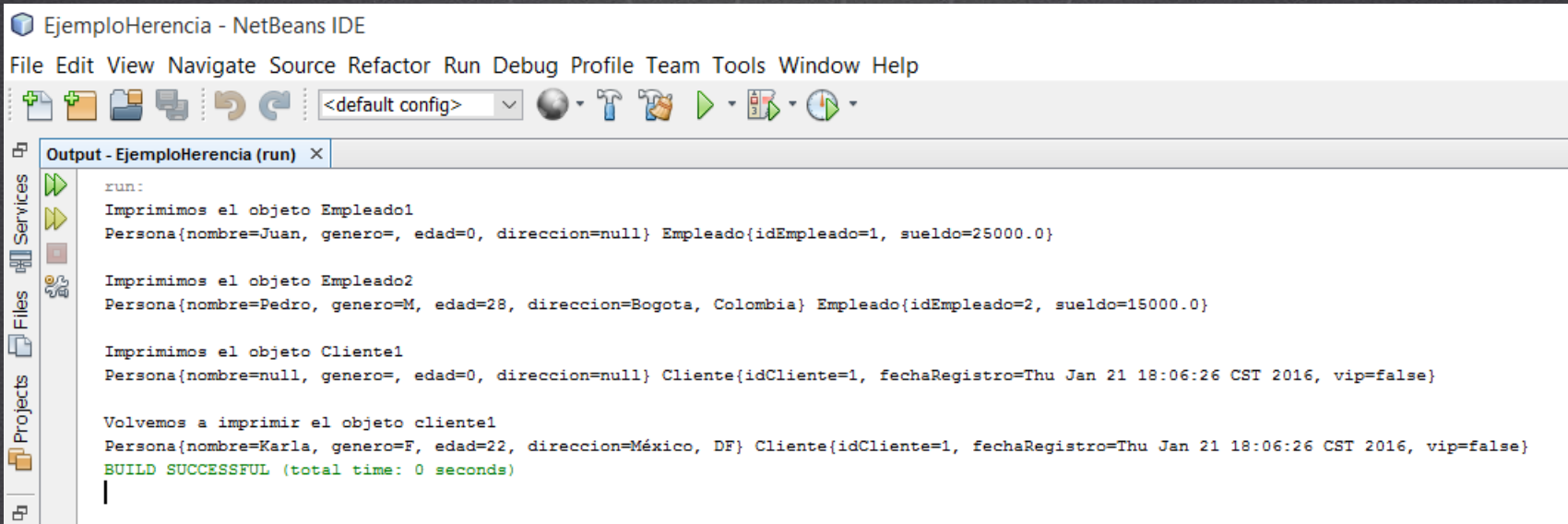
    public static void main(String[] args) {
        Empleado e1 = new Empleado("Juan", 25000);
        System.out.println("Imprimimos el objeto Empleado1");
        System.out.println(e1);

        Empleado e2 = new Empleado("Pedro", 15000);
        e2.setEdad(28);
        e2.setGenero('M');
        e2.setDireccion("Bogota, Colombia");
        System.out.println("\nImprimimos el objeto Empleado2");
        System.out.println(e2);

        //Al crear un nuevo objeto de java.util.Date nos regresa la fecha actual
        Cliente c1 = new Cliente(new java.util.Date(), false);
        System.out.println("\nImprimimos el objeto Cliente1");
        System.out.println(c1);
        //Completamos el objeto persona asociado a este objeto cliente
        c1.setNombre("Karla");
        c1.setEdad(22);
        c1.setGenero('F');
        c1.setDireccion("México, DF");
        System.out.println("\nVolvemos a imprimir el objeto cliente1");
        System.out.println(c1);
    }
}
```


PASO 3. EJECUTAMOS EL PROYECTO

Ejecutamos nuestro proyecto. El resultado es:



The screenshot shows the NetBeans IDE interface. The title bar reads 'EjemploHerencia - NetBeans IDE'. The menu bar includes 'File', 'Edit', 'View', 'Navigate', 'Source', 'Refactor', 'Run', 'Debug', 'Profile', 'Team', 'Tools', 'Window', and 'Help'. The toolbar contains icons for file operations, a configuration dropdown set to '<default config>', and a green play button. The 'Output - EjemploHerencia (run)' window is active, displaying the following text:

```
run:
Imprimimos el objeto Empleado1
Persona{nombre=Juan, genero=, edad=0, direccion=null} Empleado{idEmpleado=1, sueldo=25000.0}

Imprimimos el objeto Empleado2
Persona{nombre=Pedro, genero=M, edad=28, direccion=Bogota, Colombia} Empleado{idEmpleado=2, sueldo=15000.0}

Imprimimos el objeto Cliente1
Persona{nombre=null, genero=, edad=0, direccion=null} Cliente{idCliente=1, fechaRegistro=Thu Jan 21 18:06:26 CST 2016, vip=false}

Volvemos a imprimir el objeto cliente1
Persona{nombre=Karla, genero=F, edad=22, direccion=México, DF} Cliente{idCliente=1, fechaRegistro=Thu Jan 21 18:06:26 CST 2016, vip=false}
BUILD SUCCESSFUL (total time: 0 seconds)
```

TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Probar con distintos valores y verificar el resultado.
- Crear más variables tanto de objetos Persona, Empleado y Cliente y verificar resultados.



Experiencia y Conocimiento para tu vida

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica el concepto de Herencia y varios puntos más.
- Hemos observado que conforme avanzamos el curso comienza a ser mayor el código que escribimos y más interesante. Conforme vayamos avanzando iremos integrando este tema, el cual será de los más recurrentes al momento de trabajar con Java, ya sea de manera directa o indirecta, pero el tema de herencia está presente todo el tiempo.

CURSO ONLINE

FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx