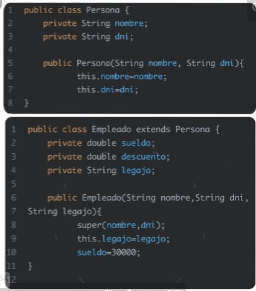
# Apuntes POO

# Herencia en JAVA

Vamos a modelar las clases que tienen nuestra super clase en nuestra clase hija. Indicamos la herencia con la famosa palabra extends en la clase hija y decir de donde se extiende esta misma.



En el caso de arriba vemos que la clase **Empleado** es una clase hija de Persona. Teniendo en cuenta que cuando creamos el constructor de nuestra clase hija, también invocamos el constructor de la clase Padre.

# Sobre Carga de métodos

La sobre carga de métodos es que tengan la misma función en este caso (nombreDeLaFuncion), pero esta misma recibe unos parámetros diferentes y un return por así decirlo diferente a la otra.

Text

Description automatically generated

Acá vemos todos métodos iguales, que se distingue uno del otro de main gracias a los parámetros que nosotros le damos al método.

Text

Description automatically generated\

Después miramos como que se da un override, ya que es NECESARIO DARLES OTRO COMPORTAMIENTO A ESTOS MISMOS.

Text

Description automatically generated

# Métodos heredados por la clase object

## String toString():

El metodo toString trata de representar con **texto** el objeto, si, este mismo se tiene que sobreescribir en la clase.

Text

Description automatically generated

Teniendo en nuestro archivo main la siguiente ejecución de este con un objeto ya creado.

Text

Description automatically generated

* boolean equals(Object o) : Lo desarrollamos para compara un objeto con otro, objeto. Este método también lo sobrescribimos en nuestro programa @***override1***. Lo hacemos con la misma firma que dice ***Object o.***
* Text

  Description automatically generated

Ya de ahí tenemos dos formas de compararlo con el ***instanceof , que lo que haces es comprobar si es una instancia de EMPLEADO***, O SEA QUE SI NO nos va a retornar un falso.

Text

Description automatically generated

La otra forma de comprar es con el .getClass(), lo que hacemos acá es comprar la clase en donde pertenecen ambos objetos.

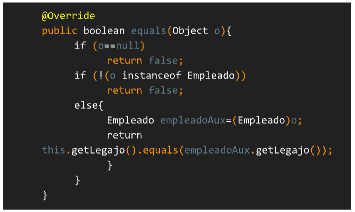
Text

Description automatically generated

Ahora después de saber que Nuestro objeto viene de la misma instancia o de la misma clase lo que hacemos es comparar los ATRIBUTOS que este posee y mirar si son iguales a los de nuestro otro OBJETO. Que sería el famoso Casting.

## Casting

Lo que hacemos acá es, crear un objeto de la clase empleado como auxiliar, después, Empleado le pedimos el legajo y este hacemos el método equals (), para compararlo con el objeto auxiliar pidiendo también su legajo.



* int hashCode(): El hash code es un método de object que lo que haces es darnos un numero único de mi atributo por así decirlo, dice que solo se trabaja con números primos este hashcode.

# EJERCICIO DE CONCLUSION NOVIEMBRE 3

Diagram

Description automatically generated