Sobrecarga y sobreescritura

DigitalHouse >
Coding School

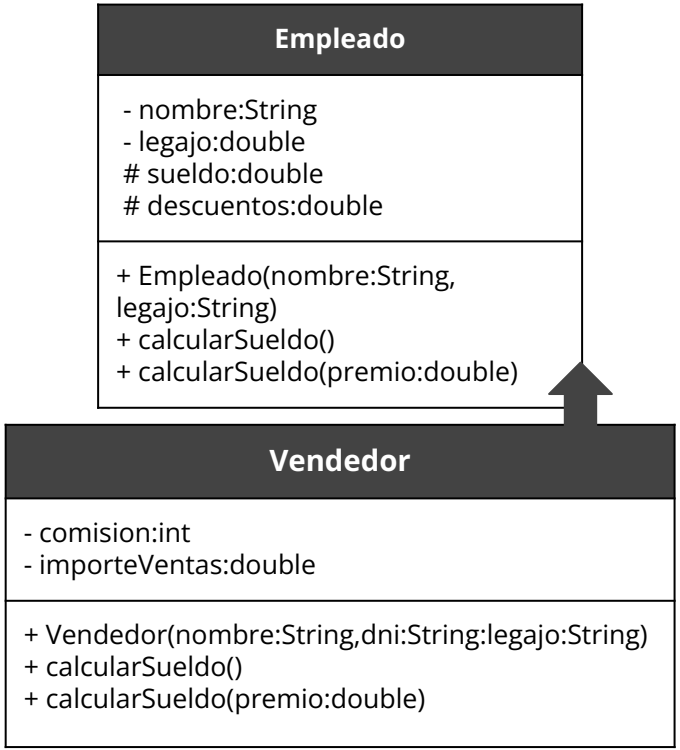
Certified Tech Developer
The Ultimate Degree

Índice

1 | Las clases

Las clases

Un empleado tiene un legajo y un sueldo. Su sueldo se puede calcular simplemente restando los descuentos al sueldo básico. Ocasionalmente, recibe un premio, por lo que el cálculo del sueldo en ese caso varía, ya que habría que sumarle este premio.
Por otro lado, hay vendedores, que también son empleados, en este caso, además de su sueldo básico, reciben una comisión por venta.



2 | Sobrecarga

Mismo nombre distinto comportamiento

Recordemos que solo podemos sobrecargar un método si varía su firma. Como vemos en este caso:

```
public class Empleado{
      private   String nombre;
      private   String legajo;
      protected double sueldo;
      protected double descuentos;

            public double calcularSueldo(){
                  return sueldo-descuentos;
            }
            public double calcularSueldo(double premio){
                  return sueldo-descuentos + premio;
            }
      }
```

Cuando se usa cada uno...

Cuando se utilizan los métodos, según los parámetros que se pasan, actuará el método cuya firma coincida con los parámetros utilizados.

```
public class Main {

   public static void main(String[] args) {

      Empleado miEmpleado = new Empleado("Juan", "1111");
      System.out.println("Sueldo a cobrar: " + miEmpleado.calcularSueldo());
      System.out.println("Sueldo a cobrar: " + miEmpleado.calcularSueldo(5000));

      }
}
```

3 | Sobreescritura

La clase Vendedor

A la clase Vendedor, con los métodos sobreescritos, es necesario darles otro comportamiento para un Vendedor. Por eso, surge la necesidad de sobreescribirlos

```
public class Vendedor extends Empleado{
	private int comision;
	private double importeVentas;

	@Override
	public double calcularSueldo(){
		return sueldo-descuentos + importeVentas/100*comision;
		}
	@Override
	public double calcularSueldo(double premio){
		return sueldo-descuentos + premio+ importeVentas/100*comision;
		}
}
```

Sobreescritura

```
@Override
public double calcularSueldo(){
      return sueldo-descuentos + importeVentas/100*comision;
      }
@Override
public double calcularSueldo(double premio){
      return sueldo-descuentos + premio+ importeVentas/100*comision;
      }
```

@Override nos indica que se anula el comportamiento anterior del método, lo estamos sobreescribiendo para darle una forma distinta de resolver. Para los objetos Vendedor, el método a ejecutar es este que sobreescribe el anterior.


DigitalHouse>
Coding School