

UD06: Javadoc



1. Exercise 1: `gestionEmpleados`

1. 1. Exercise 1:

1. 2. Exercise 2:

1. 3. Exercise 3:

2. Fuentes de información

1. Exercise 1: gestionEmpleados

Una empresa quiere hacer una gestión informatizada básica de sus empleados. Para ello, de cada empleado le interesa:

- Nombre (String)
- DNI (String)
- Año de ingreso (número entero)
- Sueldo bruto anual (número real)

2. Diseñar una clase Java `Empleado`, que contenga los atributos (privados) que caracterizan a un empleado e implemente los métodos adecuados para:

- Crear objetos de la clase: **Constructor** que reciba todos los datos del empleado a crear.
- Consultar el valor de cada uno de sus atributos. (**Consultores** o **getters**)
- `public int antiguedad()`. Devuelve el número de años transcurridos desde el ingreso del empleado en la empresa. Si el año de ingreso fuera posterior al de la fecha actual, devolverá 0. Para obtener el año actual puedes usar:

```
1 | int añoActual = Calendar.getInstance().get(Calendar.YEAR);
```

- `public void incrementarSueldo(double porcentaje)`. Incrementa el sueldo del empleado en un porcentaje dado (expresado como una cantidad real entre 0 y 100).
- `public String toString()`. Devuelve un `String` con los datos del empleado, de la siguiente forma:

```
1 | Nombre: Juan González
2 | Dni: 545646556K
3 | Año de ingreso: 1998
4 | Sueldo bruto anual: 20000 €
```

- `public boolean equals(Object o)`. Método para comprobar si dos empleados son iguales. Dos empleados se consideran iguales si tienen el mismo DNI.
- `public int compareTo(Empleado o)`. Se considera menor o mayor el empleado que tiene menor o mayor DNI (el mismo criterio que al comparar dos strings).
- Método estático `public static double calcularIRPF(double salario)`. Determina el % de IRPF que corresponde a un salario (mensual) determinado, según la siguiente tabla:

Desde salario (incluido)	Hasta salario (no incluido)	% IRPF
0	800	3
800	1000	10
1000	1500	15
1500	2100	20
2100	infinito	30

3. Diseñar una clase Java `TestEmpleado` que permita probar la clase `Empleado` y sus métodos. Para ello se desarrollará el método `main` en el que:

- Se crearán dos empleados utilizando los datos que introduzca el usuario.
- Se incrementará el sueldo un 20 % al empleado que menos cobre.
- Se incrementará el sueldo un 10% al empleado más antiguo.
- Muestra el IRPF que correspondería a cada empleado.
- Para comprobar que las operaciones se realizan correctamente, muestra los datos de los empleados tras cada operación.

4. Diseñar una clase `Empresa`, que permita almacenar el nombre de la empresa y la información de los empleados de la misma (máximo 10 empleados) en un array. Para ello, se utilizarán tres atributos: nombre, plantilla (array de empleados) y `numEmpleados` (número de empleados que tiene la empresa) En esta clase, se deben implementar los métodos:

- `public Empresa (String nombre)`. Constructor de la clase. Crea la empresa con el nombre indicado y sin empleados.
- `public void contratar(Empleado e) throws PlantillaCompletaException`. Añade el empleado indicado a la plantilla de la empresa, siempre que quepa en el array. Si no cabe, se lanzará la excepción `PlantillaCompletaException`.
- `public void despedir(Empleado e) throws ElementoNoEncontradoException`. Elimina el empleado indicado de la plantilla. Si no existe en la empresa, se lanza `ElementoNoEncontradoException`.
- `public void subirTrienio (double porcentaje)` Subir el sueldo, en el porcentaje indicado, a todos los empleados cuya antigüedad sea exactamente tres años.
- `public String toString()`. Devuelve un `String` con el nombre de la empresa y la información de todos los empleados. La información de los distintos empleados debe estar separada por saltos de línea.

5. Diseñar una clase Java `TestEmpresa` que permita probar la clase `Empresa` y sus métodos. Para ello, desarrolla el método `main` y en él ...:

- Crea una empresa, de nombre "CataDaw".
- Contrata a varios empleados (con el nombre, DNI, etc. que quieras).
- Usa el método `subirTrienio` para subir un 10% el salario de los empleados que cumplen un trienio en el año actual.
- Despide a alguno de los empleados.
- Trata de despedir a algún empleado que no exista en la empresa.
- Muestra los datos de la empresa siempre que sea necesario para comprobar que las operaciones se realizan de forma correcta.

1.1. Exercise 1:

- Generate the package `gestionEmpleados` (Unidad 5 de PRG) with all the classes, methods and properties commented with Javadoc and all the tags. At least three of the comments must include HTML.
- Export the project as ZIP and send it through AULES site to the teacher.

1.2. Exercise 2:

- Explain in a pdf document how to generate Javadoc HTML from Netbeans project.
- Add screenshots of the three methods that contains HTML in Javadoc as seen in Netbeans.
- Export the document as PDF and send it through AULES site to the teacher.

1.3. Exercise 3:

- Export the folder that contains the HTML files in your `dist` folder compressed as a ZIP and send it through AULES site to the teacher.

2. Fuentes de información

- [Wikipedia](#)
- [Uml Diagrams](#)
- [Code&Coke \(Fernando Valdeón\)](#)
- Apuntes IES El Grao (M^a Isabel Barquilla?)
- [Apuntes IOC \(Marcel García\)](#)
- [Apuntes José Luis Comesaña](#)
- [Apuntes IES Luis Vélez de Guevara 17-18 \(José Antonio Muñoz Jiménez\)](#)
- Oracle provides material on how to document program interfaces:
([How to write doc comments for the javadoc tool](#))