

Ejercicio 3

Dada las siguientes clases:

Empleado

```
package visibilidad1;
public class Empleado{
    private int sueldo;
    String nombre;
    protected String direccion;
    public int edad;
}
```

Jefe

```
package visibilidad1;
public class Jefe extends Empleado{
    public Jefe(){

    }
}
```

EmpleadoEspecial

```
package visibilidad2;
public class EmpleadoEspecial{
    public String nombreEs;
    private float sueldoEs;
    protected String direccionEs;
    int edad;
}
```

JefeEspecial

```
package visibilidad1;
import visibilidad2.EmpleadoEspecial;
public class JefeEspecial extends EmpleadoEspecial{
    public JefeEspecial(){

    }
}
```

Las clases *Empleado*, *Jefe* y *JefeEspecial* pertenecen al paquete **visibilidad1** y la clase *EmpleadoEspecial* pertenece al paquete **visibilidad2**. La clase Principal debe ir en el paquete **visibilidad1**.

Comprueba:

- 1º En la función constructora de la clase Jefe, prueba a poner "super" y ver qué es lo que te deja usar de la clase Empleado.

- 2º En la función constructora de la clase JefeEspecial, prueba a poner "super" y ver qué es lo que te deja usar de la clase EmpleadoEspecial.
- 3º Añade a la clase Jefe un nuevo atributo de tipo "EmpleadoEspecial" (recuerda que hay que importar dicha clase para poder usarla fuera de su paquete).

Instancia dicho objeto dentro de la función constructora, pon "nombre_Objeto_tipo_Empleado_Especial" y comprueba qué es lo que te deja usar.

Ejercicio 4

Escribe un programa que pida los datos de una serie de empleados y jefes que tiene la empresa "Eléctrica, SA". De cada empleado nos interesa: nombre de la empresa, nombre del empleado, edad, porcentaje de incremento de sueldo y sueldo base.

Ten en cuenta lo siguiente:

- El nombre de la empresa es el mismo para todos y es un dato que no se podrá cambiar.
- El sueldo base es el mismo para todos.
- Cada empleado gana el sueldo base más el porcentaje aplicado a cada empleado.

De los jefes, como empleados que son, nos interesan los mismos datos que antes más el plus que tiene como jefe y el nombre del departamento del que son responsables. Ten en cuenta que el plus es el mismo para todos los jefes.

El programa siempre partirá de un sueldo base de 1000€ y de un plus para los jefes de 250€.

Una vez pedidos los datos, se visualizará la información de todos ellos.

Posteriormente, el programa gestionará un menú como este:

DADO EL SIGUIENTE MENÚ:

1. Modificar el sueldo base de todos los empleados.
2. Modificar el plus de todos los jefes.
3. Visualizar los datos de todos los empleados.
0. Salir.

ELIJA UNA OPCIÓN:

Ejercicio 5

Para este ejercicio se utilizarán las clases del Ejemplo6.

- a. En la clase Jefe sobrescribe el método **pedirDatos()** de la clase **Empleado**. Comprueba qué sucede.
- b. ¿Podrías explicar por qué si llamas a un método desde un constructor, el compilador te aconseja una de las siguientes opciones?

1. Que la clase, donde está el método que llamamos dentro del constructor, sea **final**.
2. Que el método al que llamas, dentro del constructor, sea **final**.
3. Que el método al que llamas, dentro del constructor, sea **estático**.
4. Que el método al que llamas, dentro del constructor, sea **private**.

Ejercicio 6

Crea una clase que sea abstracta y lleva a cabo las siguientes comprobaciones:

1. Inserta métodos no abstractos.
2. Inserta métodos abstractos.
3. Pon cuerpo a un método abstracto.
4. Declara un objeto con ese tipo de clase.
5. Instancia un objeto de dicha clase.
6. Declara un atributo abstracto.
7. Crea una nueva clase que no sea abstracta y prueba a insertar un método que sea abstracto.

Haz que esa clase herede de la clase abstracta y comprueba que te obliga a insertar los métodos abstractos que tiene dicha clase.

Ejercicio 7

Dada la clase Empleado siguiente:

```
final class Empleado{  
    String nombre;  
    int edad;  
  
    final void pedirDatos(){}  
    void visualDatos(){}  
}
```

1. Crea la clase Jefe que herede de la clase Empleado. ¿Qué ocurre?
2. Quita el modificador "**final**" a la clase **Empleado**. Haz que la clase Jefe herede de la clase Empleado y añade a la clase Jefe un método con esta cabecera: **void pedirDatos()**. ¿Qué ocurre?