



## Práctica 6. Herencia

El objetivo de la siguiente práctica es consolidar los conocimientos adquiridos sobre herencia en Java, además de ponerlos en práctica. Para ello, se propone realizar los siguientes ejercicios:

### 1. Desarrolla las siguientes:

- a. **Clase Empleado**, que se caracteriza por poseer los atributos: nombre, apellido1, apellido2, nif, edad y salario.
- b. **Clase Comercial**, que se caracteriza por ser una especialización de Empleado, y además tener el atributo comisión. Asegúrate que de esta clase nunca se podrán crear subclases.
- c. **Clase Repartidor**, que se caracteriza por ser una especialización de Empleado, y además tener el atributo zona. Asegúrate que de esta clase nunca se podrán crear subclases.
- d. Crea para todas las clases sus correspondientes métodos (**constructores**, **constructor copia**, **getters/setters**).
- e. Crea un método que se llamará **mostrarAtributos** en todas las clases, y que se encargará de mostrar en cada una de las clases los atributos del objeto. Este método tendrá el mismo nombre en todas las clases (override).
- f. Crea un método que se llamará **pedirAlta** en todas las clases, y que se encargará de pedir al usuario para cada una de las clases los atributos del objeto. Este método tendrá el mismo nombre en todas las clases (override).
- g. Crea un **programa principal**, y mediante un **menú**, realice el alta de empleados, comerciales o repartidores (utiliza el **método pedirAlta**). Los empleados, comerciales o repartidores se guardarán en una lista en el programa principal.
- h. Además, el **programa principal**, ofrecerá en el menú la opción de **mostrarAtributos** de cualquiera de los tipos de empleado.
- i. Además, realiza un control de las excepciones que se pueden dar en la entrada por teclado de los usuarios.
- j. Utiliza siempre los métodos getters y setters cuando desees acceder a uno de los atributos de la clase.
- k. Cuando se introduzca el nombre o apellido se debe comprobar que la primera letra debe ser en mayúsculas siempre.
- l. Cuando se introduzca la edad, siempre debe ser al menos de 16 años de edad.

2. Desarrolla las siguientes clases:

- a. **Clase obra**, que se caracteriza por tener poseer los atributos: nombre, fecha de creación, id de la obra (el id se autoincrementará y será común a todas las obras) y una lista de los artistas que la crearon.
- b. **Clase Artista**, que se caracteriza por tener el nombre, apellidos, año de nacimiento, lugar de nacimiento. Asegúrate que de esta clase nunca se podrán crear subclases.
- c. **Clase Pinturas**, que se caracteriza por ser una especialización de obra, y que además incorpora los atributos alto, ancho y precio subasta.
- d. **Clase Musical**, que se caracteriza por ser una especialización de obra, y que además incorpora los atributos duración, nº de canciones, precio cd y precio vinilo.
- e. **Clase Audiovisual**, que se caracteriza por ser una especialización de obra, y que además incorpora los atributos duración, precio dvd y precio bluray.
- f. Crea un método que se llamará **mostrarAtributos** en todas las clases, y que se encargará de mostrar en cada una de las clases los atributos del objeto. Este método tendrá el mismo nombre en todas las clases (override).
- g. Crea un método que se llamará **pedirAlta** en todas las clases, y que se encargará de pedir al usuario para cada una de las clases los atributos del objeto. Este método tendrá el mismo nombre en todas las clases (override).
- h. Crea un **programa principal**, y mediante un **menú**, realice el alta de obras, artista, pinturas, musical y audiovisual (utiliza el **método pedirAlta**). Las obras se guardarán en una lista en el programa principal. Este método nos debemos asegurar que no se realizará ninguna sobreescritura.
- i. Además, el **programa principal**, ofrecerá en el menú la opción de **mostrarAtributos** de cualquiera de los tipos de obras.
- j. Crea el método **localizar\_Obras\_Artista**, en el que se pedirá para un nombre de artista, sus obras disponibles y toda la información relativa a ellas. Debe encontrar dada una cadena, cualquier obra que la contenga, por ejemplo: Para Miguel, podría encontrar "miguel, Miguel, Miguel Ángel, Miguel Ángel). que encuentre
- k. Además, realiza un control de las excepciones que se pueden dar en la entrada por teclado de los usuarios.
- l. Utiliza **siempre** los métodos getters y setters cuando desees acceder a uno de los atributos de la clase.
- m. Valida que las obras musicales deben contener más de una canción.
- n. La fecha de creación debe ser entre el año 0 y el día de hoy (investiga).
- o. Realiza los casos test utilizando JUNIT de todos aquellos métodos en los que no se realiza ninguna petición por teclado al usuario.