

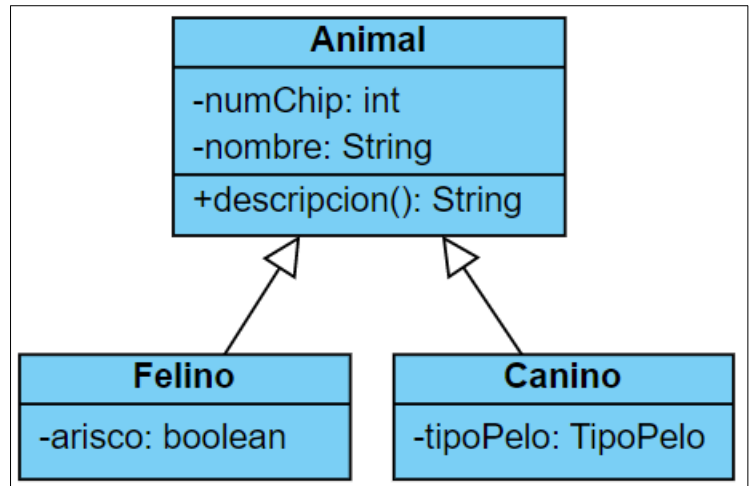
# Ejercicios de Herencia en P00



## Relación de ejercicios

**Ejercicio 1.** Implementa el siguiente conjunto de clases: Animal, Felino y Canino. Debes tener en cuenta las siguientes características:

- Todo Animal tiene un número de chip (entero) y un nombre (cadena de texto).
- Todo Animal debe tener un método descripción que indique su número de chip y su nombre.
- Todo Felino tiene número de chip, nombre y si es arisco o no.



- Si el Felino es arisco, en su descripción debe aparecer la descripción de cualquier Animal junto con los símbolos: (!).
- Todo Canino tiene número de chip, nombre y tipo de pelo. El tipo de pelo únicamente puede ser (uno de ellos): duro, rizado, corto o largo.
- En la descripción de un Canino debe aparecer el número de chip, un guion, el tipo de pelo entre paréntesis y seguidamente el nombre del Animal.
- Todas las clases deben contar con su constructor con parámetros y con los *getters* y *setter* correspondientes.

**Ejercicio 2.** Implementa un programa que genere una colección de Felinos y Caninos (todos en la misma colección). Posteriormente, muestra por pantalla la información de todos ellos.

**Ejercicio 3.** En una tienda se venden 2 tipos de ordenadores: portátiles y de sobremesa. Ambos tipos de ordenadores se caracterizan por su código (cadena de texto) y por su precio. Por defecto, todo ordenador tendrá un eslogan genérico con el texto “Calidad al mejor precio”. Además, los ordenadores portátiles tienen un atributo peso y los de sobremesa la descripción del tipo de torre.

Implementa las clases, con sus atributos y métodos, teniendo en cuenta que el código no se debe poder modificar una vez creado el objeto. Cada uno de los tipos de ordenadores (portátil o sobremesa) tendrán que tener su propio eslogan que modifique el genérico.

**Ejercicio 4.** Partiendo del ejercicio anterior, implementa un programa que instancie un ordenador portátil y un ordenador de sobremesa para realizar las siguientes acciones.

- Modificar el peso del portátil.
- Modificar el precio del portátil, aplicando un descuento del 10%.
- Visualizar todos los datos del portátil por pantalla.
- Visualizar el código junto con el eslogan.
- Modificar el tipo de torre y visualizar todos los datos del ordenador de sobremesa por pantalla.
- Modificar el precio del ordenador de sobremesa, incrementando el precio un 20%.
- Realizar un método que nos permita recibir una lista de ordenadores y aplicar el descuento que le pasemos a todos los ordenadores. Decide el mejor sitio para ponerlo. El descuento tiene que ser superior al 0% y no llegar al 100%.

**Ejercicio 5.** Implementa un programa que tenga la superclase Profesor y la subclase ProfesorEmerito. Las características son las siguientes:

- Profesor tiene nombre (cadena de texto), edad (entero) y anyosConsolidados (entero) que deben ser privados pero su acceso y modificación debe ser protegido, es decir, no se permitirá su manipulación pública pero sí desde la propia clase, subclase o paquete.
- Profesor tendrá un método privado denominado *presentarse()* que devolverá una cadena de texto formada por el nombre y edad del profesor.
- Profesor tendrá un método público *cumplirAnyos()* que incrementará en una unidad la edad. Este método no se debe poder sobrescribir.
- ProfesorEmerito tiene un atributo privado llamado *anyosEmerito* cuya manipulación (*get* & *set*) también debe ser privada.

- ProfesorEmerito tendrá un método público *getSalarioBase()* que obtendrá el salario base como:  $(925 + \text{anyosConsolidados} * 33.25 + 47.80 * \text{anyosEmerito})$
- De la clase ProfesorEmerito no se debe poder heredar.

Posteriormente, implementa una clase denominada Claustro (en el mismo paquete) que contenga una colección de Profesor. Además, debe tener un método público, llamado *presentarProfesores()*, que devuelva un texto con el nombre y la edad de todos los profesores del claustro.

Tras implementar lo anterior, genera un programa (en otro paquete) que:

- ◆ Cree un objeto de la clase ProfesorEmerito y llame al método *presentarse()*. Si no funciona, razona el motivo.
- ◆ Cree una lista de Profesores con un Profesor y un ProfesorEmerito. Con dicha lista...:
  - Crea un objeto de la clase Claustro pasándole la lista de Profesores recientemente creada y llame a su método *presentarProfesores()*.
  - Tenga un método que reciba la lista de Profesores y muestre el nombre y años consolidados de todos junto con su salario base. Quizá tengas algún error en tiempo de compilación... Razona el motivo. No tienes que solucionarlo, simplemente razonar el motivo por el que ocurre el error (si es que se produce alguno).

**Ejercicio 6.** Amplía [este ejercicio](#) con un método que dada una colección de Animales, únicamente muestre la información de los Felinos que son ariscos.

**Ejercicio 7.** Amplía el ejercicio anterior con un método que se le pase si queremos mostrar los Felinos o los Caninos y muestre la información únicamente del tipo de Animal que se le haya pasado por parámetro.