

Ejercicio 1

Para la realización de esta práctica **se creará un archivo .java** llamado **figuras** con las clases necesarias en cada uno de los ejercicios **y otro .java con un programa** de prueba llamado **apli_figuras**.

Implementar las siguientes clases dentro de **figuras**:

- **Clase cuadrado** (padre), que contiene un atributo, lado, y un **método CalculaArea**. lado1 x lado1.
- **Clase rectángulo** (hereda de cuadrado), que añade otro lado y **sobrescribe el método CalculaArea (@Override)**. lado1 x lado2.
- **Clase cubo** (hereda de cuadrado), que no añade atributos, pero también **sobrescribe el método CalculaArea(@Override)**. lado1 x lado1 x 6;

Crea un objeto de cada clase y llama a cada uno de los métodos. Para distinguir cuál se está ejecutando podéis poner un mensaje en pantalla al principio de cada método CalculaArea como el siguiente: `System.out.println("Soy el método CalculaArea de la clase...")` y luego que haga el return.

Ejercicio 2

Define la **clase MiJefe** con atributos: nombre, telefonoOficina (**protected**), telefonoCasa (**private**) y telefonoMovil darle valores a los atributos en su creación y un **método MuestraTelefonos** que permita visualizar todos los atributos.

Crea una subclase empleado1 que herede de MiJefe, con **otro método MuestraTelefonos** que visualice los datos del Jefe y del Empleado.

Ejercicio 3

Se desea representar puntos geométricos en un espacio de tres dimensiones.

Crear la **clase Punto3D**, El constructor de la clase deberá aceptar los tres valores, coordenadas del punto. Las operaciones que deseamos que ejecute un objeto de la clase son:

- Obtener las tres componentes correspondientes a un punto cualquiera.
- Obtener la distancia del punto al centro de coordenadas (0,0,0).

La solución al problema consiste en extender el comportamiento de la clase punto.

Dentro de la clase heredada, para poder hacer referencia a los atributos y métodos heredados de la superclase, se deberá hacer uso de la palabra reservada **super**.

<pre>class Punto { protected double x; private double y; public Punto() {} public Punto(double x, double y) { this.x=x; this.y=y; } double ordenada () { return y; } } // cierra la clase Punto</pre>	<pre>class Punto3d extends Punto { ... } //cierra la clase</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------

Crear una clase llamada **pruebaPunto3D** para comprobar su funcionamiento.

Ejercicio 4

Realizar el siguiente ejercicio partimos de un clase:

Clase vehiculo con atributos numRuedas entero por defecto 4 numPuertas entero por defecto 4 y llevaRadio booleano por defecto true.

A parte del constructor por defecto necesitamos un constructor para el numRuedas y numPuertas en el cual tambien nos diga por pantalla que es un vehiculo de “x” puertas e “y” ruedas.

Incorporará un **metodo llamado conducir** que nos informará que estamos conduciendo y si la radio esta en marcha.

Crear una **subclase llamada coche**, que en su constructor por defecto nos indique que es un coche.

Tendremos un **metodo llamado cierraPuertas** que nos dirá por pantalla “Cierro las puertas”

El metodo conducir lo sobreescribiremos para añadir el metodo **cierraPuertas**.

Crear **una subclase llamada moto**, que en su constructor por defecto nos pondrá 0 puertas y 2 ruedas sin radio y nos dirá que “Soy una moto”.

Para acabar en la **clase publica VehiculoTest** estará nuestro main en el contruiremos un coche y posteriormente se le pasará el metodo conducir a dicho coche.

Lo mismo con una moto, comprobar y entender los resultados.

Ejercicio 5

Partimos de una **clase abstracta llamada Animal** con los atributos especie, ruido y comida todos ellos String.

Al constructor le pasaremos las tres cadenas de texto.

Tendremos **dos metodos para ObtenerComida y ObtenerEspecie** con sus respectivos return.

Crear la **subclase Perro** con el atributo patas entero,

A su constructor se le pasarán los cuatro parámetros.

Crear los metodos **ObtenerComida** y **ObtenerRuido** para el caso del perro.

Crear la **subclase Gato** con el atributo edad entero.

A su constructor se le pasarán los cuatro parámetros.

Crear los metodos **ObtenerComida** y **ObtenerRuido** para el caso del gato.

Esta clase tambien tendrá un **metodo llamado Ver** el cual devolverá por pantalla el ruido del animal la edad que tiene y su comida.

Crear la clase pública llamada granja en la que estará el main de la aplicación y tenemos que crear un perro ("Galgo", "Guau", "Hueso", 4) y un gato ("Siames", "Miau", "Sardina", 4)

Para acabar hemos de mostrar por pantalla el ruido del perro y su comida, y el metodo Ver para el caso del gato.