

Ejercicio 1.

Definir una clase que represente a un **coche**. En la definición se debe incluir:

- el modelo
- el color
- si la pintura es metalizada o no
- la matrícula
- El tipo de coche que puede ser MINI, UTILITARIO, FAMILIAR o DEPORTIVO
- El año de fabricación
- la modalidad del seguro, que puede ser a terceros o a todo riesgo

Añadir a la clase Coche del ejercicio un método de nombre **imprimeCoche** que imprima el modelo y el color del coche.

Ejercicio 2.

Escribe un programa que disponga de una clase para representar las **asignaturas** de una carrera:

- Una asignatura tiene un nombre, un código numérico y el curso en el cual se imparte.

Los valores iniciales han de proporcionarse en el constructor.

La clase ha de tener métodos para obtener los valores de los atributos.

El programa ha de construir un objeto con los siguientes valores:
nombre "Matemáticas", código 1017, curso 1.

A continuación, el programa ha de imprimir los valores del objeto por pantalla

Ejercicio 3.

Definir una clase llamada **PUNTO** que represente un punto en un espacio bidimensional.

La clase debe disponer de un constructor con las coordenadas del punto y métodos accesorios a las coordenadas

Añade a la clase Punto un constructor sin parámetros, que permita construir puntos en el origen de coordenadas

Escribe un programa que instancie cuatro puntos: el primero situado en el origen, el segundo situado en (5,3), el tercero en (2, -1) y el cuarto estará situado en el punto medio entre el segundo y el tercero.

Ejercicio 4.

Escribe una clase que represente un **reloj** que señale la hora, el minuto y el segundo.

La clase dispondrá de dos constructores, uno sin parámetros que pone el reloj a 0:0:0 y otro al que se le pasa la hora, los minutos y los segundos.

Se proporcionarán los siguientes métodos:

Uno que da la hora, los minutos y los segundos, separados por el carácter ":" en una cadena.

Otro que también da la hora pero en formato 24 horas (como el anterior) o en formato 12, en cuyo caso debe distinguir entre "am" (para las horas de 0 a 11) o "pm" (para las horas de 12 a 23), también en una cadena.

Un método para poner el reloj en hora. Se le pasa la hora y los minutos, poniendo los segundos a 0.

Un método para poner el reloj en hora al que, además, se le pasan los segundos.

Ejercicio 5.

Una empresa dispone de **3 almacenes** para contenedores, cuyas capacidades respectivamente son: 2, 4 y 8.

Dichos almacenes se encuentran en distintas ubicaciones el 1º en la propia empresa distancia 0 km, el 2º a 5 km y el 3º a 10 km.

Los camioneros llegan a la empresa con el objetivo de cargar o descargar un contenedor.

Desarrolla una aplicación, que de forma automática diga a los camioneros el almacén más cercano al que deban dirigirse, minimizando los costes de combustible y tiempo.

El camionero indicará al programa si va a:

1. Cargar un contenedor
2. Descargar un contenedor

El programa les indicará el almacén al que debe dirigirse.

El programa se creará utilizando una **clase** que represente el **almacén**. La **clase principal** creará los **almacenes** y **determinará a que almacén debe de dirigirse**.

Ejercicio 6

Una **compañía** pretende una gestión informatizada de sus empleados. Para ello, de cada **empleado** le interesa lo siguiente.

Nombre (String)

DNI (String)

Estado Civil (carácter)

Año de ingreso (número entero)

Sueldo bruto anual (número real)

Diseña una **clase empleado** que contenga los atributos que caracterizan a un empleado e implemente los métodos adecuados para:

- Crear objetos de la clase.
- Consultar el valor de cada uno de los atributos.
- Incrementar el sueldo del empleado un 10%. Llamar al método **incrementarSueldo**
- Devolver por pantalla los atributos de un empleado, con el formato que se considere adecuado. Llamar al metodo **toString**

Diseñar una **clase** Java llamada **Compañía**, que permita almacenar la información de los empleados (máximo 10) en un array. En esta clase, se deben implementar los métodos que permitan:

- Añadir un nuevo empleado, ha de preguntar al final si ¿Desea meter un empleado? S/N . **Método meterEmpleado.**
- Incrementar en un 10% el sueldo de aquellos empleados que lleven 10 ó más años en la compañía. **Método incrementarSueldo**
- Crear una String correspondiente al listado de todos los empleados (método toString). La información de los distintos empleados debe estar separada por el carácter de salto de línea. **Método listarEmpleados**