

## 0485.- Programación.

[Área personal](#) / [Mis cursos](#) / [Programación](#) / [U5 - POO. Herencia. Interfaces.](#) / [U5 - POO. Herencia. Interfaces \(19/20 - Mañana\).](#)

### U5 - POO. Herencia. Interfaces (19/20 - Mañana)

#### EJERCICIO 1

Queremos gestionar información relativa a los vehículos que puede albergar un garaje.

Deberemos hacer todas esas clases en el mismo paquete. Los atributos de las clases sólo podrán ser accedidos desde la misma clase.

El garaje contiene vehículos, **en general**, de los que se guarda el nombre del vehículo (Ej. "La Dorada", "La perla negra", etc) y el número de personas que puede transportar. Estos vehículos, pueden ser a su vez, de tipo Aéreo y Terrestre. De los vehículos Aéreos, puede haber Helicópteros y Avionetas. De los vehículos Terrestres puede haber Coches y Motocicletas.

Los Coches y las Motocicletas guardarán la matrícula (Ej. "9875PLJ"), el año de fabricación y el color (que pueden ser AZUL, AMARILLO, ROJO Y VERDE).

Implementa un constructor y los métodos toString() y equals() (este último basado en la matrícula). Implementa la interfaz Comparable para estas clases con un criterio de ordenación basado también en la matrícula). Implementar, asimismo, un comparador para hacer ordenaciones basadas en el año de fabricación.

Todos los vehículos podrán añadir personas para cada viaje, por lo que tendrán un método *transportar* que lo permita.

Los Helicópteros y las Avionetas guardarán la altitud a la que están volando, así como el código OACI (Convenio sobre Aviación Civil Internacional), como por ejemplo "UP-A300I". Obviamente, podrán volar. Implementa esta función que devuelva una cadena que indique el tipo de aeronave que es (Helicóptero o Avioneta) y la altitud a la que está volando. Implementa un constructor que añada el código OACI e implementa el método toString() que imprima el tipo de aeronave y dicho código.

Realiza un programa principal que realice las siguientes tareas:

- Crear 3 coches, 3 motos, un helicóptero y una avioneta.
- Los datos, obviamente, serán inventados.
- Mostrar ordenados los 3 coches y las 3 motos atendiendo a todos los criterios solicitados.
- Añadir 9 personas al helicóptero.
- Mostrar el contenido del helicóptero.

**NOTA: Recordad que debéis probar las distintas clases en un Main para asegurarnos que todo está correcto.**

#### Instrucciones para la Entrega

Entregar un **archivo comprimido ( zip o rar)** que contenga cada uno de los ficheros correspondientes a los ejercicios (EjX.java siendo X el número del ejercicio)

EL fichero comprimido se llamará **Apellido1\_Apellido2\_U5\_T1\_Entrega.zip** (o rar)

Evita ñ y acentos en el nombre.

**Deberan subirse a [gitHub](#)** los ficheros en la carpeta correspondiente (U5\_T1) que se encontrará dentro de la carpeta U5

**NOTA: Para cada uno de los ejercicios se bajará un 20% del mismo por cada error sintáctico corregido por el profesor para que el ejercicio pudiera funcionar correctamente. Se tendrá también en cuenta la eficiencia de los algoritmos.**

### Estado de la entrega

Estado de la entrega	No entregado
----------------------	--------------