

PRUEBA EVALUABLE UD 5 – 05/03/2021

Camion	Flota
- matricula: texto	- flota: conjunto de Camion
- revisionPasada: lógico	+ Flota ()
- depositoCompleto: lógico	+ altaCamion(Camion c)
- kmRecorridos: entero	+ contieneCamion(Camion c) devuelve lógico
+ constructor con todas las propiedades	+ bajaCamion(Camion c) devuelve lógico
+ llenaDeposito()	+ imprimeFlota()
+ realizaRuta()	+ ponerFlotaEnRuta()
+ equals + hashCode + toString	

La clase *Camion* representa a un camión de una empresa de transporte y tendrá las siguientes restricciones:

- El constructor recibe todos los parámetros para inicializar todas las propiedades.
- El método *llenaDeposito* simplemente establece la propiedad correspondiente a true
- El método *realizaRuta* sumará 500 km a los kilómetros recorridos del camión siempre que este haya pasado la revisión y tenga el depósito lleno. Si no había pasado la revisión se lanzará una *RevisionPendienteException* con un mensaje descriptivo. Si no tenía el depósito lleno se lanzará una *DepositoIncompletoException* con un mensaje descriptivo.
- Debes crear los métodos *toString*, *equals* y *hashCode*

La clase *Flota* representa una flota de camiones y tendrá las siguientes restricciones:

- El constructor debe inicializar el conjunto y asociarlo a la propiedad *flota*.
- El método *altaCamion* dará de alta un camión en la flota, añadiéndolo a la misma.
- El método *bajaCamion* eliminará el camión que se recibe como parámetro de la flota y devolverá un lógico que indique si consiguió realizar la operación o no.
- El método *contieneCamion* devolverá un lógico que indique si el camión que se recibe como parámetro pertenece a la flota o no.
- El método *imprimeFlota* utilizará un iterador para recorrer la flota de camiones y los imprimirá con el siguiente formato: "N – toString del camión" donde N será un número entero creciente (1, 2, 3... Ver ejemplo de ejecución en la siguiente hoja).
- El método *ponerFlotaEnRuta* recorrerá la flota de camiones y llamará al método *realizaRuta* de cada camión. Se deben tratar todos los camiones de la flota, independientemente de que algunos puedan provocar excepciones y otros no.
 - Si un camión se puede poner en ruta se imprimirá "En ruta: + toString del camión".
 - Si se produjo una excepción se mostrará "Motivo: + toString del camión" (ver ejemplo de ejecución en la siguiente hoja) donde el motivo podrá ser "Pendiente revisión" o bien "Depósito incompleto".

Por último, se debe crear una clase *PruebaFlota* que añada 5 camiones a una flota, uno de los cuales no tendrá el depósito lleno y otro no habrá pasado la revisión. A continuación, se imprimirá la flota y se llamará al método *ponerFlotaEnRuta*.

Ejemplo de ejecución:

```
1 - Camion{matricula=4567, revisionPasada=false, depositoCompleto=true, kmRecorridos=1500}
2 - Camion{matricula=5678, revisionPasada=true, depositoCompleto=true, kmRecorridos=1500}
3 - Camion{matricula=1234, revisionPasada=true, depositoCompleto=true, kmRecorridos=0}
4 - Camion{matricula=2345, revisionPasada=true, depositoCompleto=true, kmRecorridos=500}
5 - Camion{matricula=3456, revisionPasada=true, depositoCompleto=false, kmRecorridos=1000}
-----
Pendiente revisión: Camion{matricula=4567, revisionPasada=false, depositoCompleto=true, kmRecorridos=1500}
En ruta: Camion{matricula=5678, revisionPasada=true, depositoCompleto=true, kmRecorridos=2000}
En ruta: Camion{matricula=1234, revisionPasada=true, depositoCompleto=true, kmRecorridos=500}
En ruta: Camion{matricula=2345, revisionPasada=true, depositoCompleto=true, kmRecorridos=1000}
Depósito incompleto: Camion{matricula=3456, revisionPasada=true, depositoCompleto=false, kmRecorridos=1000}
```