

1. Crear una **Aplicación Java** que:

- **Defina** el interfaz **IVehiculo**, **requisito funcional** que debe cumplir toda clase que quiera ser un vehículo, que declare los métodos:
  - **obtenerNumeroRevisiones**
  - **mostrarInfoRevisiones**
- **Defina** la **clase abstracta** **JavaBean Vehiculo**, **contenedora de la lógica común** a todas las clases que quieran ser un vehículo, que:
  - Implemente **PERO NO CODIFIQUE** el interfaz **IVehiculo**:
    - A partir de los kilómetros recorridos por el vehículo, el método **obtenerRevisiones** debe calcular y devolver el número de revisiones realizadas al vehículo teniendo en cuenta que se realiza una revisión **cada cierto número de km que dependerá del tipo de vehículo**.
    - El método **mostrarInfo** debe mostrar por pantalla, de forma conveniente:
      - El tipo de vehículo.
      - Los kilómetros actuales del vehículo.
      - El número de revisiones realizadas al vehículo.
      - En caso de que la revisión del vehículo esté próxima debe indicarse cuántos kilómetros faltan para la misma.
  - **Defina y encapsule** de forma que **no pueda heredarse**, un atributo llamado **contador** que contenga el número de kilómetros que lleva recorridos el vehículo y que no debe ser negativo. El cuentakilómetros de todo vehículo nuevo nunca está exactamente a 0 sino que marca una pequeña cantidad aleatoria de kilómetros rodados, en general siempre inferior a un **margen que depende del tipo de vehículo**, necesaria para realizar el control de calidad.
  - **Defina** sendos constructores **predeterminado** y **parametrizado**.
  - **Defina** el método **avisarRevision** que devuelve **true** si faltan 50 km o menos para la siguiente revisión y **false** en caso contrario. Este método **no se puede heredar**.
- **Defina** la clase **JavaBean Camion** que:
  - Se derive de la clase **Vehiculo**.
  - Implemente los métodos abstractos.
  - Debe realizar las revisiones cada 5000 km.
  - El margen de kilometraje compra es inferior a los 20 km.
  - El margen de proximidad de revisión está en 50 km.
- **Defina** la clase **JavaBean Moto** que:
  - Se derive de la clase **Vehiculo**.
  - Implemente los métodos abstractos.
  - Debe realizar las revisiones cada 12500 km.
  - El margen de kilometraje compra es inferior a los 5 km.
  - El margen de proximidad de revisión está en 50 km.

- **Defina** la clase de librería llamada **UtilesVehiculo** que contenga aquellos recursos de programación que se puedan extraer de las clases anteriores, tales como:
  - El método **validarContador** que devuelve **true** si el valor del atributo **contador** cumple con los requisitos del enunciado y **false** en caso contrario.
  - El método **obtenerInicial** que devuelve un valor aleatorio positivo que representa los km con los que sale un vehículo de fábrica. Cada tipo de vehículo tiene un margen diferente.
  - El método **obtenerRestante** que devuelve los kilómetros que le restan al vehículo para tener que pasar una nueva revisión.
  - El método **comprobarProximidad** que devuelve **true** si faltan pocos kilómetros para que el vehículo tenga que pasar la revisión y **false** en caso contrario.
- Cree **sendos objetos** de la clase **Camion** utilizando sus **respectivos constructores** y por cada uno de ellos:
  - Mostrar la información relativa a las revisiones del vehículo llamando a su método **mostrarInfoRevisiones**.
  - Modificar el cuentakilómetros del objeto llamando al correspondiente **método mutador**.
  - Mostrar de nuevo la información relativa a las revisiones del vehículo llamando a su método **mostrarInfoRevisiones**.
- Cree **sendos objetos** de la clase **Moto** utilizando sus **respectivos constructores** y por cada uno de ellos:
  - Mostrar la información relativa a las revisiones del vehículo llamando a su método **mostrarInfoRevisiones**.
  - Modificar el cuentakilómetros del objeto llamando al correspondiente **método mutador**.
  - Mostrar de nuevo la información relativa a las revisiones del vehículo llamando a su método **mostrarInfoRevisiones**.