

Enunciado:

Desarrollar un sistema de gestión de un garaje siguiendo las siguientes especificaciones.

En el garaje se cambian las ruedas tanto de coches como de motos. El precio del cambio de una rueda se fija al abrir el garaje, al igual que la capacidad máxima de vehículos, ya sean motos o coches. Si no hubiese lugar disponible para registrar un vehículo nuevo, habrá que tener contemplado un sistema de aviso para quien esté dejando su vehículo.

El sistema de gestión del garaje requiere un mecanismo para ingresar y retirar los vehículos, conocer el número total de vehículos recibidos en ese momento, el precio que supondría cambiar todas las ruedas de todos los vehículos, y el kilometraje medio de todos ellos.

La información que manejaremos de los coches entre otras cosas es la marca y el número de puertas. Mientras que de las motos será la marca y la cilindrada.

Las clases relacionadas con los vehículos se guardarán en el paquete `vehiculos` mientras que las clases `Garaje` y `GarajeTest` se guardarán en el paquete `sistemaGaraje`.

La clase `GarajeTest` tiene un método `main` en el que se crea un `Garaje`, una serie de vehículos que se irán recibiendo en el garaje y por último se imprimirá por pantalla toda la información general del garaje así como la información de cada vehículo.

Actividades:

1. Realizar el análisis del requerimiento determinando el dominio del problema.
2. Identificar los objetos y las clases del sistema.
3. Determinar las relaciones entre clases (asociación, composición, Herencia).
4. Documentar el análisis detallando cada paso realizado.
5. Realizar el diagrama de casos de uso para modelar las interacciones entre los usuarios de un sistema y sus casos de uso.
6. Realizar un diagrama de clases que permita modelar las clases en el sistema que representa el funcionamiento de un Garaje y sus interrelaciones.
7. Mapear el diagrama de clases a código Java.
8. El programa debe ser 100% orientado a objetos, incluyendo en la codificación comentarios que detallen dónde se ve representado cada uno de los pilares del paradigma de la orientación a objetos.
9. Se deben respetar tanto las reglas sintácticas como la reglas por convención del lenguaje Java.

10. El código debe estar organizado de manera prolija, legible y correctamente indentado.

En la fecha estipulada se deberá entregar:

- a) La documentación UML de los diagramas solicitados.
- b) El detalle de los pasos necesarios para el análisis y diseño del aplicativo de forma escrita.
- c) El código fuente óptimo para importar.