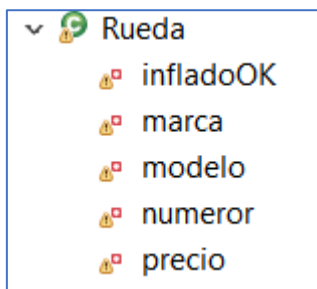


## Ejercicio extra de ArrayList (Coche):

### Coche:

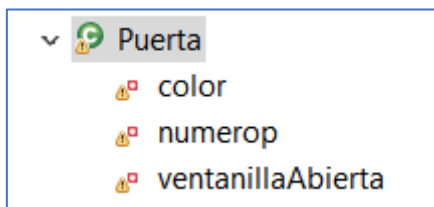
1. Queremos implementar el software de un coche, en concreto las ruedas y las puertas. Crea las siguientes clases:

**Rueda-> Guarda la información relacionada con una rueda del coche**



(añade los métodos getter, setter y constructores).

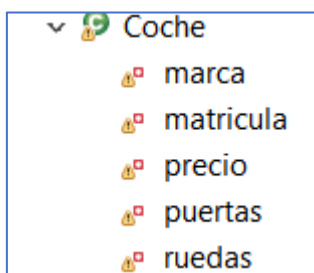
**Puerta-> Guarda la información relacionada con una puerta del coche**



(añade los métodos getter, setter y constructores).

**Coche -> guarda la información del coche.**

- ✓ Las ruedas se guardan en el ArrayList ruedas
- ✓ Las puertas se guardan en el ArrayList puertas



(añade los métodos getter, setter y constructores).

## Ejercicio extra de ArrayList (Coche):

Realiza el siguiente programa principal.

Crea un coche con todos sus datos, incluidas las puertas y las ruedas.

A continuación, muestra el siguiente menú:

- |   |
|---|
| <ol style="list-style-type: none"><li>1) <b>Mostrar datos del coche</b></li><li>2) <b>Inflar ruedas</b></li><li>3) <b>Mostrar precio total de las ruedas</b></li><li>4) <b>Cambiar puerta</b></li><li>5) <b>Quitar puertas de un color:</b></li><li>6) <b>Salir</b></li></ol> |
|---|

**Mostrar datos del coche:** Muestra todos los datos del coche, incluidas todas las puertas

**Inflar Ruedas:** Inflará todas las ruedas del coche (atributo infladoOK a true en todas las puertas)

**Mostrar precio total de las ruedas:** Muestra el precio total de las puertas (suma el precio de todas las ruedas)

**Cambiar puerta:** Elige un numero de puerta y cámbiala por otra (pide los datos por teclado).

**Quitar puertas de un color:** Pide un color y elimina las puertas del coche que coincida con el color especificado.

---