

Hoja de ejercicios 43_44:

- **Clases y objetos.**
- **Atributos y métodos. Uso de *this* para atributos.**
- **Constructores.**

43. Crea una clase llamada *Vehículo*:

- Con los **atributos privados** *numeroRuedas*, *velocidadMaxima*, *velocidadActual* y *peso*.
- Implementa los **métodos** necesarios para acceder a estos atributos de manera **pública** (*getters* y *setters*).
- La clase dispondrá de un **constructor** que necesitará como parámetros los valores iniciales para todos sus atributos.
- Crea un **método** público *boolean esIgual(Vehiculo)* que sirva para comparar dos vehículos, de manera que devuelva *true* o *false* dependiendo de si son iguales o no (tienen todos sus atributos el mismo valor o no). El método recibirá como parámetro un objeto de la clase *Vehículo*.
- Crea un **método** público void *copia(Vehiculo)* que copiará los atributos de un vehículo en otro. El método recibirá como parámetro un objeto de la clase *Vehículo* del cual se copiarán sus valores.
- Crea una clase aparte con el método *main* para probar las funcionalidades de la clase *Vehículo*.

44. Crea una clase llamada *Nombres*, capaz de gestionar una lista de nombres de un tamaño determinado:

- El **constructor** recibirá como parámetro el número máximo de nombres que albergará.
- Tendrá los siguientes **métodos**:
 - *boolean anadir(String)*: Añade a la lista el nombre pasado como parámetro. Devuelve *false* si la lista está llena o si ya existe el nombre, *true* si se añade con éxito. **No admite repetidos**.
 - *boolean eliminar(String)*: Elimina de la lista el nombre pasado como parámetro. Devuelve *true* en caso de éxito y *false* si no encuentra el nombre a eliminar.
 - *void vaciar()*: Elimina todos los nombres de la lista.
 - *String mostrar(int)*: Devuelve el nombre que se encuentra en la posición pasada como parámetro (la primera es la posición 0). Devuelve *null* si no es una posición válida.
 - *int numNombres()*: Devuelve el número de nombres que hay actualmente.
 - *int maxNombres()*: Devuelve el número máximo de nombres que puede albergar.
 - *boolean estaLlena()*: Devuelve *true* si la lista está llena y *false* en caso contrario.
- Realiza el ejercicio 39 de nuevo, pero utilizando la clase *Nombres*.