

## Hoja de ejercicios 52\_53:

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

52.- 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 que devuelva *boolean*, llamado *esIgual(Vehiculo vehiculo)* que sirva para comparar dos vehículos, de manera que devuelva *true* o *false* dependiendo de si son iguales o no (se consideran iguales si el valor de todos sus atributos es el mismo). El método recibirá como argumento un objeto de la clase *Vehículo*.
- Crea un método público que devuelva *void* llamado *copia(Vehiculo 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*.

53.- Crea una clase llamada *Nombres*, capaz de gestionar una lista de nombres, sin repetir, de un tamaño determinado:

- El constructor recibirá como parámetro el número máximo de nombres que albergará.
- Tendrá los siguientes métodos:
  - *boolean anadir(String nombre)*: 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 nombre)*: 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 posicion)*: 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.

54.- Utilizando la clase *Nombres* recién creada, rehaz el ejercicio 46.