



Gestión de Garaje		
1º DAW - ETS		
Departamento de Informática	Fecha	Luke Eric Marten Llorente

Descripción

Se desarrollará un sistema de gestión para un garaje mediante dos clases: **Coche** y **Garaje**. La clase **Coche** representará información sobre un vehículo, mientras que **Garaje** permitirá administrar una colección de coches. También se implementará la clase **Main_Garaje** para probar su funcionamiento.

Requisitos

Clase **Coche**

Propiedades

- **String marca**: Marca del coche.
- **String modelo**: Modelo del coche.
- **String matricula**: Matrícula del coche.
- **int añoFabricacion**: Año de fabricación.

Métodos

- **Constructores**:
 - Uno que inicializa todos los atributos.
 - Otro que inicializa marca, modelo y añoFabricacion, asignando "No definido" a la matrícula.
- **Getters y Setters** para todos los atributos.
- **toString()**: Devuelve una cadena con la información del coche en el formato:
 - "Marca: [marca] - Modelo: [modelo] - Matrícula: [matricula] - Año: [añoFabricacion]".

Clase **Garaje**

Propiedades

- `Coche[] coches`: Array que almacena los coches en el garaje.
- `int[] plaza`: Números de plaza asignados a cada coche.

Métodos

- **Constructores**:
 - Uno que inicializa un garaje vacío.
 - Otro que recibe un array de `Coche`, lo almacena y asigna números de plaza consecutivos comenzando desde 1.
- `void append(Coche nuevoCoche)`: Agrega un coche nuevo al garaje.
- `toString()`: Devuelve la lista de coches en formato alineado:
 - "Plaza: [plaza] - Marca: [marca] - Modelo: [modelo] - Matrícula: [matricula] - Año: [añoFabricacion]".

Clase **Main_Garaje**

- Crear instancias de `Coche` con distintos valores.
- Inicializar un array de coches y usarlo para crear un `Garaje`.
- Agregar coches adicionales con `append()`.
- Imprimir el garaje con `toString()`.