

## Ejercicios con clases y array de objetos

### Ejercicio 1

Implementa una clase **Motocicleta** que represente a una motocicleta. De una motocicleta interesa conocer los caballos de potencia (cv), la velocidad punta en km/h, las tres letras de su matrícula, y el número de la matrícula.

Además de definir los atributos necesarios para representar las características mencionadas, habrá que implementar:

- Un constructor por defecto y otro que reciba el valor de todos los atributos.
- Los métodos que permitan acceder a dichos atributos (get) y los que permitan modificarlos recibiendo el nuevo atributo como parámetro (set).
- Un método **mostrarMatricula()** que muestre la matrícula al completo por pantalla.  
Ejemplo: 1234DFR.
- Un método **main** en el que se creen 2 motocicletas, la primera con el constructor por defecto, y después se introduzcan sus valores mediante los métodos set. Otra mediante el constructor con parámetros. Finalmente se mostrarán por pantalla los datos de las dos motocicletas con los métodos get y el método mostrarMatricula()

**Mejora:** Modifica la clase Motocicleta para que el número y las letras de la matrícula se utilicen y almacenen como un solo String siguiendo el formato de las matriculas.

### Ejercicio 2

Vamos a implementar las estructuras necesarias para evaluar un módulo de un ciclo cualquiera. Para ello implementa:

- 1) Una clase **Alumno** con los siguientes atributos: **nombre, dni y la calificación de cada trimestre.**

Además de definir los atributos necesarios para representar las características mencionadas, habrá que implementar:

- a) Un constructor por defecto y un constructor que reciba el valor del nombre y dni, el resto se inicializarán por defecto. El valor por defecto de cada trimestre será -1 para indicar que este trimestre aún no se ha evaluado.
- b) Getters y Setters.

- c) Un método **calificaciónFinal** que devuelva la media ponderada de los tres trimestres. En el caso de que algún trimestre aún no haya sido evaluado, entonces se devolverá un -1 para indicar que aún no se puede evaluar.
- d) Un método **informeAlumno** que muestre por pantalla todos los datos del alumno: nombre, dni, calificación de los tres trimestres y calificación final.
- 2) Una clase módulo con los siguientes atributos: nombre, curso, profesor y un array con al menos 5 alumnos.
- Además de definir los atributos necesarios para representar las características mencionadas, habrá que implementar:
- Constructor con argumentos.
  - Getters y Setters
  - Un método para **calcularMediaTrimestre** que recibe como parámetro el número de trimestre. Si aún no ha sido evaluado el trimestre, entonces se devolverá un -1 para indicar que aún no se puede calcular.
  - Un método **informeCurso** que mostrará por pantalla todos los datos de los alumnos: nombre, dni, calificación de los tres trimestres y calificación final.
- 3) Una clase moduloTest en la que tendrás que:
- Crear 5 alumnos
  - Crear un módulo
  - Calcular la media de los tres trimestres para ese módulo
  - Mostrar un informe final del curso.
  -

## Ejercicio 3

Implementa una clase **Carta** que represente a una carta de una baraja de póker. De una carta interesa conocer su número y su palo (almacenado como una única letra).

- Un constructor por defecto y un constructor de copia.
- Los métodos get que permitan acceder a los atributos.
- Los métodos set que en lugar de recibir los nuevos valores como argumento, los soliciten por teclado. Se comprobarán que sean válidos y se volverán a solicitar mientras no lo sean. Para que el número de la carta sea válido deberá estar entre 1 y 13 y el palo deberá ser una de las siguientes letras: **c,p,r,t** que corresponden a corazones, picas, rombos y tréboles.

- Un método **mostrarCarta()** que muestre por pantalla de que carta se trata con el siguiente formato: es el <numero> de <palo> donde el número y el palo corresponden a los atributos de la carta con las siguientes particularidades:
  - El palo del número deberá mostrarse al completo, es decir se deberá sustituir la letra del palo por el palo en cuestión.
  - Si el número de la carta está entre 2 y 10 se mostrará normalmente. Para el resto de los números 1,11,12,13 se cambiará respectivamente por As, Jack, Reina, Rey .
- Un método **comparar(Carta)** que reciba una carta como argumento y la compare con la actual Indicando por pantalla cuál de ellas es la ganadora. Para esto se comprobará en primer el número, si son distintos ganará la mayor, si son iguales se compararán los palos, el orden de victoria es Trébol, Rombos, Picas, Corazones. Si las cartas son iguales se indicará que hay un empate.
- Un método **main** en el que se creen 2 cartas, la primera con el constructor por defecto y se inicialice con los métodos set, la segunda se copiará con el constructor de copia. Después se mostrarán ambas cartas con el método mostrarCarta() y se compararán con el método comparar. Después de esto se modificará la segunda con los métodos set y se volverán a mostrar y comparar.

## Ejercicio 4

Crea una clase **Libro** con los atributos **titulo**, **autor**, **isbn**, **editorial** y **precio**. Crea un array de objetos Libro y ordénalos por precio de menor a mayor. Luego, muestra el listado de libros ordenados.

## Ejercicio 5

Crea una clase **Coche** con atributos **marca**, **modelo** y **año**. Luego, crea un array de objetos Coche y realiza una búsqueda para encontrar un coche por su marca. Imprime el coche encontrado o un mensaje si no se encuentra.

## Ejercicio 6

Desarrollar una lista de Libros ordenada por título. La funcionalidad de la lista será la habitual: conocer el número de libros que hay en la lista, insertar un nuevo libro (en la posición que le corresponda), eliminar el libro de una determinada posición y obtener el libro de una determinada posición. También incluirá un método para buscar un libro a partir de una parte de su título (sin distinguir entre mayúsculas y minúsculas); el método devolverá la posición en la que se encuentra el libro (-1 si no se encuentra).