

EJERCICIOS I: DESARROLLO DE CLASES

Desarrolla las siguientes clases con el lenguaje de programación Orientado a Objetos Java:

- Motocicleta
 - Alumno
 - Carta
-

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

- Un constructor por **defecto** y otro que reciba por parámetros el valor de todos los atributos.
- Los métodos **getter y setter** para cada atributo
- El método **toString()**
- Un método público **String matricula()** que devuelva la matrícula al completo. Ejemplo: 1234DFR
- Un método publico void **mostrarMatricula()** que muestre la matrícula al completo por pantalla. Ejemplo: 1234DFR.
- Un método **main** en el que se crean 2 motocicletas, la primera con el constructor por defecto, y después se introducen sus valores mediante los métodos set. La segunda se crea con el constructor con parámetros. Finalmente se mostrarán por pantalla los datos de las dos motocicletas con los métodos **toString()**, los métodos **get** y el método **mostrarMatricula()**

Mejora: Copia la clase motocicleta con el nombre Moto para que el número y las letras de la matrícula se utilicen y almacenen como un solo String.

2. Implementa una clase **Alumno** que representa a los alumnos de una asignatura. De cada alumno interesa conocer su **dni**, su **edad** y su **nota de cada trimestre**.

- Un constructor por defecto y un constructor que reciba el valor del dni y la edad, 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.
- Los métodos **getter y setter** para cada atributo
- El método **toString()**
- Un método público **notaFinal** que devuelva la media 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.
- Un método público **informeAlumno** que muestre por pantalla todos los datos del alumno: DNI, edad, notas de los tres trimestres y nota final.
- Un método **main** en el que se crean 2 alumnos, uno por defecto y añadiéndole todos los valores con los métodos set, y otro alumno con el constructor con parámetros. Finalmente se mostrará el Informe de los dos alumnos.

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 copias.
- Los métodos que permitan acceder a dichos atributos (getters).
- Los métodos setters 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.
- El método toString()
- Un método público **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 privado **comparar(Carta carta)** que reciba una carta como argumento y la compare con la actual y devuelva -1 si esta carta gana a la carta pasada por parámetro. Devuelve 0 si hay empate y 1 si la carta pasada como parámetro gana a esta carta.

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, Rombo, Picas, Corazones. Si las cartas son iguales se indicará que hay un empate.

- Escribe un método público **mostrarGanadora(Carta carta)** que llame al método a comparar anterior.
- 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 copias. Después se mostrarán ambas cartas con el método **mostrarCarta()** y se mostrará la ganadora con el método **mostrarGanadora(Carta carta)**. Después de esto se modificará la segunda con los métodos set y se volverán a mostrar cada una para finalmente mostrar la ganadora.