

CarCity

Debido a la creciente contaminación atmosférica, el ayuntamiento de la ciudad de CarCity ha decidido limitar la circulación de vehículos por sus calles. El programa simulará la gestión de los vehículos que pueden circular por la ciudad en un día determinado, según su matrícula, año de matriculación y si son eléctricos o no.

Se utilizarán las siguientes clases:

Clase Matricula

Atributos de la clase: *numeros* (String), *letras* (String)

Todas las matrículas están formadas por 4 números seguidos de 3 letras. El atributo *numeros* contendrá la parte numérica de la matrícula y el atributo *letras* contendrá las letras. Todos los atributos de la clase son privados.

Escribe en la clase los métodos que consideres necesarios: Constructor o constructores adecuados, Setters y Getters, *toString()*, etc.

Clase Coche

Atributos de la clase: *matricula* (de tipo *Matricula*), *añoMatriculacion* (int), *marca* (String), *eléctrico* (boolean). Todos los atributos de la clase son privados.

Escribe en la clase los métodos que consideres necesarios: Constructor o constructores adecuados, Setters y Getters, *toString()*, etc.

Clase Menu

El programa utilizará la siguiente clase Menu:

```
public class Menu {

    private int opcion;

    public void mostrar() {
        System.out.println(" 1. Nuevo coche");
        System.out.println(" 2. Mostrar todos los coches");
        System.out.println(" 3. Mostrar los que pueden circular hoy");
        System.out.println(" 0. FIN");
    }

    public int leer() {
        Scanner sc = new Scanner(System.in);
        do {
            System.out.print("Introduzca opción: ");
            opcion = sc.nextInt();
        } while (opcion < 0 || opcion > 3);
        sc.nextLine(); //limpiar el intro
        return opcion;
    }

}
```

Clase Principal

Los coches se guardan en un ArrayList. Mediante un menú se controlarán las acciones a realizar.

Métodos que se deben ejecutar correspondientes a cada opción del menú:

Opción 1: método **nuevoCoche()**. Añade un nuevo coche al ArrayList coches. Se piden por teclado todos los datos del nuevo coche y se añade al ArrayList. No puede haber coches repetidos. Si se intenta añadir un coche existente se mostrará un mensaje y se volverá al menú principal. Consideramos que dos coches son iguales si tienen la misma matrícula.

Tampoco se pueden añadir coches cuyo año de matriculación sea mayor que el año actual. Si el año de matriculación no es válido se vuelve a pedir hasta que sea correcto.

También se debe comprobar que la matrícula sea correcta. Las matrículas están formadas por 4 números seguidos de 3 letras. Si la matrícula no es correcta se muestra un mensaje indicándolo y se vuelve a pedir.

Opción 2: método **mostrarTodos()** Muestra todos los coches ordenados por año de matriculación **de mayor a menor**. Si no hay coches se muestra un mensaje indicándolo.

Opción 3: método **circulanHoy()**. Muestra los coches que pueden circular hoy ordenados alfabéticamente por marca. Los coches de la misma marca aparecerán ordenados por año de matriculación de menor a mayor. Para decidir si un coche puede circular o no se debe tener en cuenta lo siguiente:

- Los coches eléctricos y los matriculados este año pueden circular todos los días.
- Para los coches no eléctricos: si el día de hoy es par pueden circular los coches cuya matrícula comienza por una cifra par. Si el día es impar circularán los coches cuya matrícula comienza por una cifra impar. Consideramos el 0 como cifra par.