GESTIÓN DEL APARCAMIENTO DE UN AEROPUERTO

Crear un programa para gestionar el aparcamiento de un aeropuerto que ofrece un servicio de recogida y entrega del coche en la terminal.

PARTE 1:
Se necesitan las siguientes clases:
≡ Valida es una interfaz que contiene un método boolean valida (String cadena); que devuelve verdadero si la cadena es correcta o falso si no lo es.

Esta interfaz se debe implementar tanto en Vehículo para comprobar que la matrícula es correcta como en el DNI de la persona (8 números y una letra)

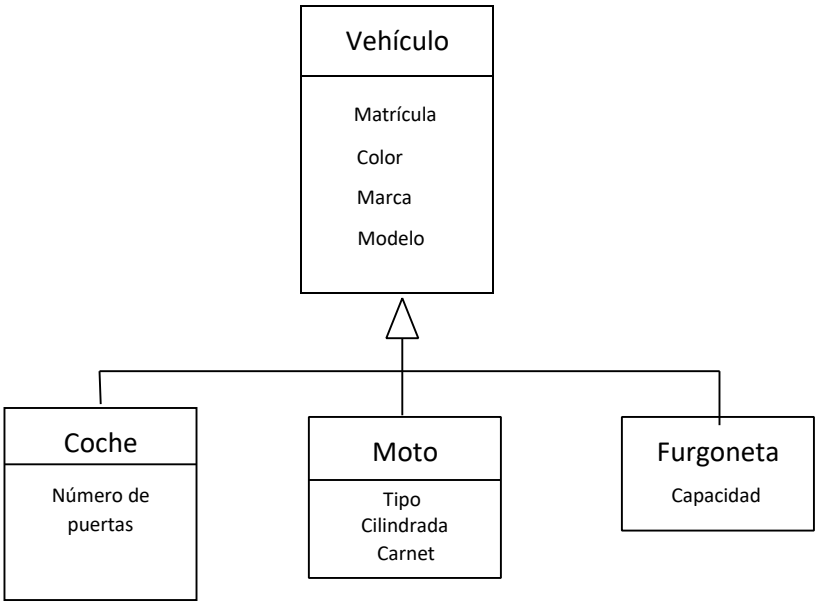
Nota: Para calcular la letra del dni hay que dividir el número del DNI entre 23 y el resto corresponde a una posición de un array con el siguiente contenido.

Resto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Letra	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Vehículo es una clase abstracta de la que heredan las clases coche, moto y furgoneta. Los atributos de estas 3 clases deben ser privados.
Dependiendo del tipo de vehículo es precio del estacionamiento se calculará de la siguiente manera:

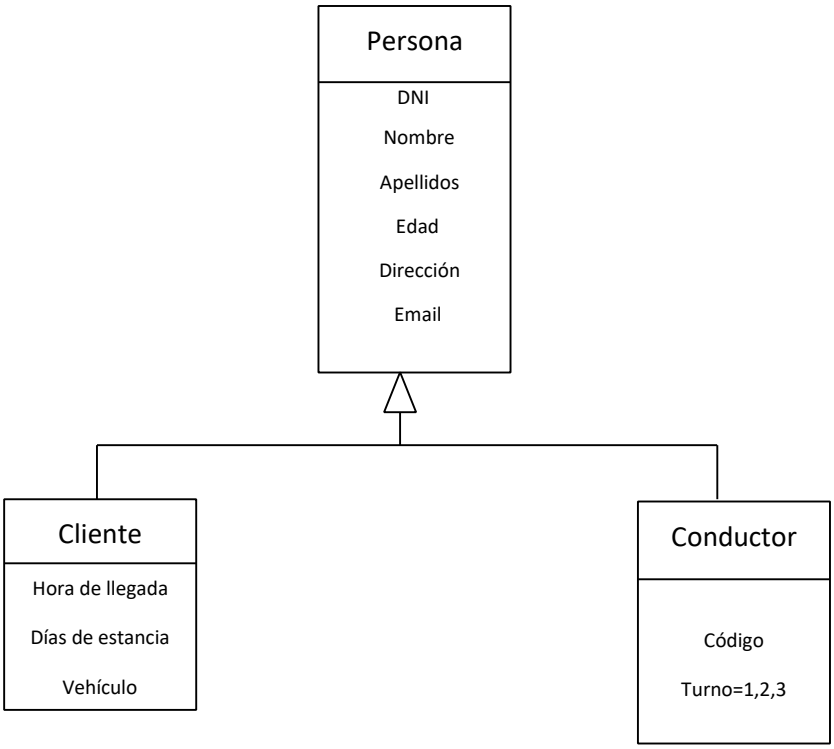
- ∴ Si es **coche** el precio será de 10,00€ por día
- ∴ Si es **moto** el precio será de 5,00€ por día
- ∴ Si es **furgoneta** el precio será de 15€ por día

La matrícula del coche debe tener 4 dígitos, un guion y 3 letras en mayúscula si no, no será considerada válida. Crea la jerarquía de clases, los constructores con y sin parámetros, los getters y setters, el método de la interfaz valida(String cadena) para la matrícula y el toString().



Crea un programa de prueba para comprobar que el funcionamiento de las clases es el correcto.

PARTE 2:
A continuación, existe la siguiente herencia de clases:



- La clase Persona también es abstracta y de ella heredan Cliente y Conductor. Implementa el método boolean valida(String cadena) para que no se almacenen DNIs incorrectos
- Para cada cliente se necesita saber el vehículo que se va a recoger, la hora de llegada y los días que ha estado en el aparcamiento.
- El conductor tendrá un código de identificación y un turno que podrá tomar los valores 1,2 y 3 correspondiendo a mañana, tarde y noche.
- Existe un archivo llamado conductor.csv que contiene los datos de los conductores que van a trabajar el día 19 de marzo.

Crea un programa llamado Aparcamiento que al iniciarse se muestre el siguiente menú y permita elegir una opción hasta que se pulse "0". Cuando se seleccione una opción se ejecutará lo que se describe a continuación:

---

1. **Leer conductores**: Lee los conductores del fichero y los guarda en un ArrayList de conductores
2. **Dar de alta cliente**: Pide los datos al usuario y lo añade al ArrayList llamado clientesaparcamiento.
3. **Guardar clientes**: Guarda los objetos de la clase cliente en un fichero de objetos llamado clientes.par
4. **Leer clientes**: Cargará los clientes de fichero clientes.par en el ArrayList de Clientes llamado clientesaparcamiento (avisar al usuario de que se inicializará de nuevo el array de clientes perdiendo los datos anteriores que no haya guardado)

   Imprimir por pantalla los clientes.

5. **Generar Tickets**: Para cada cliente genera un fichero TicketNombreApellido.txt con el importe que el cliente va a pagar según su vehículo y el número de días que ha estado en el aparcamiento.
6. **Imprimir turno de entrega**: Para cada conductor muestra los vehículos y los datos de los clientes con los que va a trabajar ese día. Para imprimir estos turnos se debe considerar que el turno 1 es de 8 a 15horas, el turno 2 es de 16 a 24 (0 horas) y el turno 3 de 1 a 8 de la mañana.