Reto 1 – Alquiler de Autos

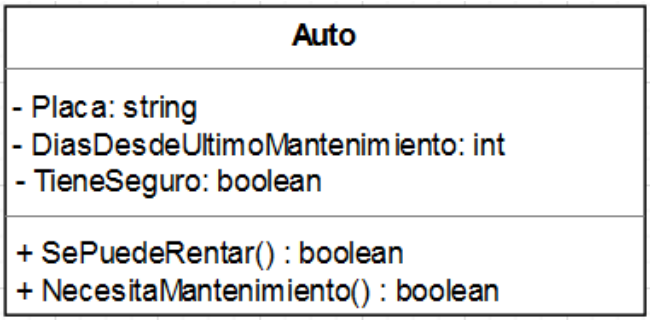
Objetivo:

El objetivo de este reto es que el estudiante reconozca y aplique los elementos básicos del paradigma de la programación orientada a objetos en un escenario abstraído de la cotidianidad.

Contexto:

Una empresa de alquileres de autos en Antioquia se está preparando para la temporada de turismo y ha hecho una inversión en varios sistemas que le facilite a los empleados hacer control sobre los vehículos que tienen disponibles. Para esto la empresa necesita que el sistema en el que puedan consultar si un vehículo en específico necesita mantenimiento y si pueden rentarlo.

En el sistema actual un **Auto** está definido como se muestra en el diagrama:



Reto:

De acuerdo a las políticas de la empresa, a todos los vehículos se les debe hacer mantenimiento cada 20 días, la función **NecesitaMantenimiento** debe retornar verdadero en caso de que hayan pasado más de 20 días desde el último mantenimiento, de lo contrario retorna falso.

La función **SePuedeRentar** valida si un vehículo puede ser alquilado y retorna verdadero si cumple con las siguientes condiciones:

- El auto tiene seguro, es decir, la propiedad **TieneSeguro** de la instancia de **Auto** tiene como valor Verdadero.
- El auto **NO** necesita mantenimiento.
- En caso de que cualquiera de estas condiciones no se cumpla debe retornar falso.

Casos de Prueba:

Para validar el correcto funcionamiento del programa considere los siguientes escenarios:

Caso de Prueba	Datos de Entrada	Salida Esperada
1. NecesitaMantenimiento()	**Auto** - Placa: "ADK847" - DiasDesdeUltimoMantenimiento: 25 - TieneSeguro: True	True
2. SePuedeRentar()	**Auto** - Placa: "GYZ175" - DiasDesdeUltimoMantenimiento: 2 - TieneSeguro: False	False
3. SePuedeRentar()	**Auto** - Placa: "FAT324" - DiasDesdeUltimoMantenimiento: 20 - TieneSeguro: True	True
4. SePuedeRentar()	**Auto** - Placa: "ADK847" - DiasDesdeUltimoMantenimiento: 25 - TieneSeguro: True	False

ENTREGA:

1. El archivo que suba a la plataforma para su calificación debe llamarse **exactamente** "Auto.java", de lo contrario no se calificará.
2. Los nombres de las clases, miembros dato y funciones deben llamarse **exactamente** como se muestra en el diagrama mostrado al comienzo del reto, la firma de su clase debe ser cómo se muestra en la siguiente imagen:

```java
public class Auto {
    private String Placa;
    private int DiasDesdeUltimoMantenimiento;
    private boolean TieneSeguro;

    public boolean NecesitaMantenimiento(){
        //Implementación
    }

    public boolean SePuedeRentar(){
        //Implementación
    }
}
```