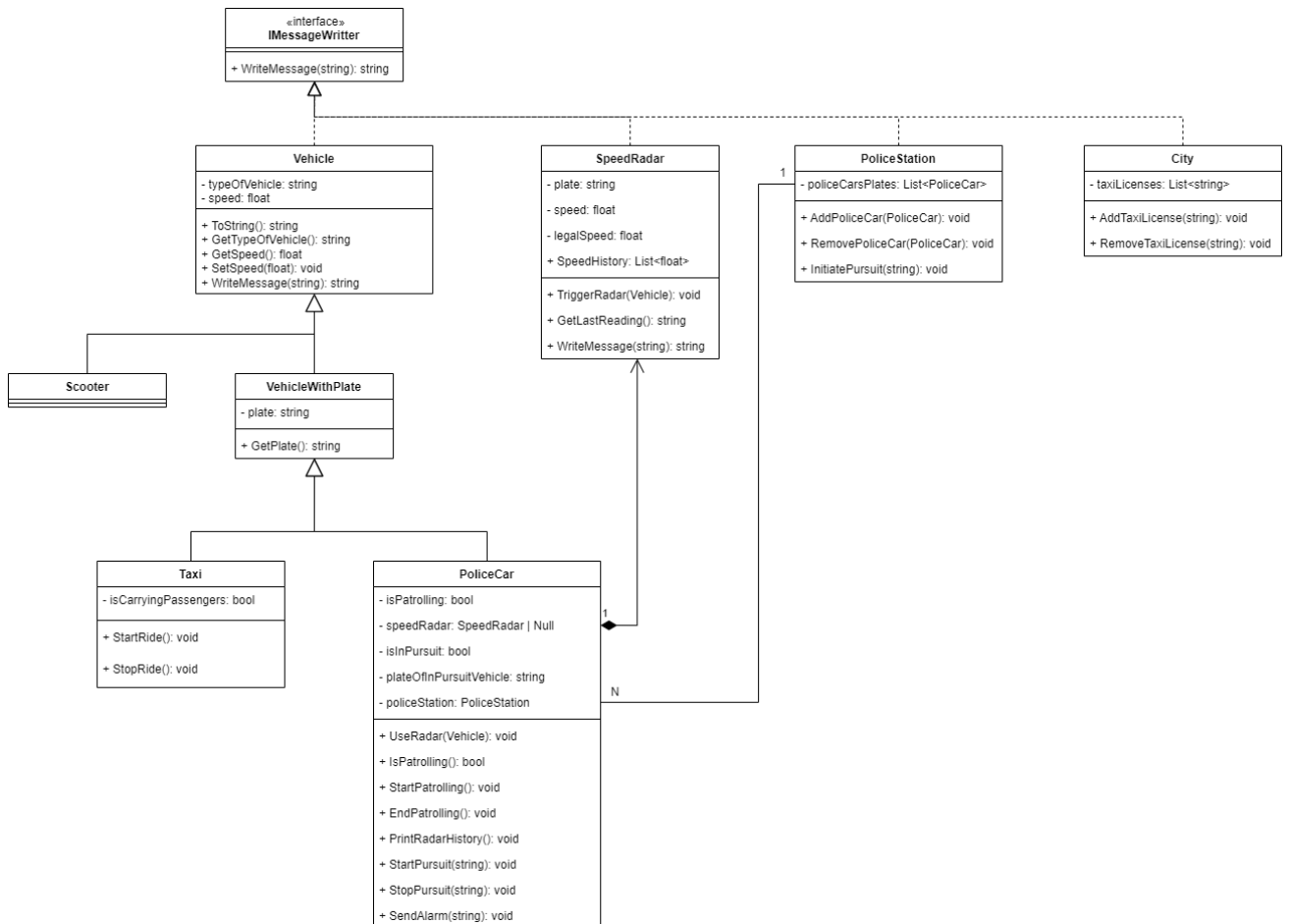


Práctica 2 - Paradigmas de la computación

Alberto Velasco Rodríguez

3 de octubre de 2024

Diagrama UML



Principios SOLID

Single responsibility principle

Cada clase tiene claramente definido su rol en el programa y los objetivos y responsabilidades asignadas.

Open/Close principle

Un ejemplo de la buena implementación de este principio es la diferencia entre la clase PoliceStation y City. En el caso de PoliceStation necesitamos poder llamar a los métodos de cada PoliceCar para iniciar una persecución, pero en el caso de City con una lista de strings de las placas de los taxis es suficiente para mantener la funcionalidad que se nos pide. De esta forma mantenemos el acceso a otros objetos al mínimo indispensable.

Liskov Substitution Principle

Para cumplir este principio se ha tenido que crear la clase VehicleWithPlate. De esta forma Scotter hereda de Vehicle y no altera el comportamiento de este al no tener matrícula.

Interface Segregation Principle

Sólo se ha implementado una interfaz y es necesaria de la forma que se ha desarrollado para comunicar los resultados por terminal de forma eficiente.

Dependency Inversion Principle

Todas las clases mantienen una correcta comunicación y paridad de nivel entre ellas.

Implementación aparato de medida

Para permitir intercambiar entre el tipo de aparato de medida que usará el PoliceCar se deberá usar una clase Meter que será lo que verá el PoliceCar y de ella heredarán el SpeedRadar y la clase que represente el nuevo alcoholímetro a implementar. Al igual que el SpeedRadar ve Vehicle no ve a Taxi en concreto.