

Prog. Orientada a Objetos

Programador full-stack

Ejercicios

Reglas para la resolución de todos los ejercicios que están a continuación:

- Resolver en un nuevo proyecto NPM.
- En los casos que corresponde armar diagramas usando draw.io y bajarlos como imagen.
- Implementar en TypeScript, separando cada clase en un archivo (relacionar con sentencias export / import).
- Subir el código y las imágenes a Github.
- Realizar pull-request para revisión.

Ejercicio I

Inciso 1

- Diseñar una clase para manejo de archivos de texto. Estos almacenan datos en forma de distintas líneas de texto con separadores particulares para registros y campos. (Diagrama)
- Implementar en TypeScript la clase del punto anterior. (utilizar librería fs de @types/node)

Inciso 2

- Utilizar la clase del ejercicio 1, como librería, para guardar y recuperar la información creada en el ejemplo de clase elegida (el ejemplo de objeto que pensó). Implementar en TypeScript.

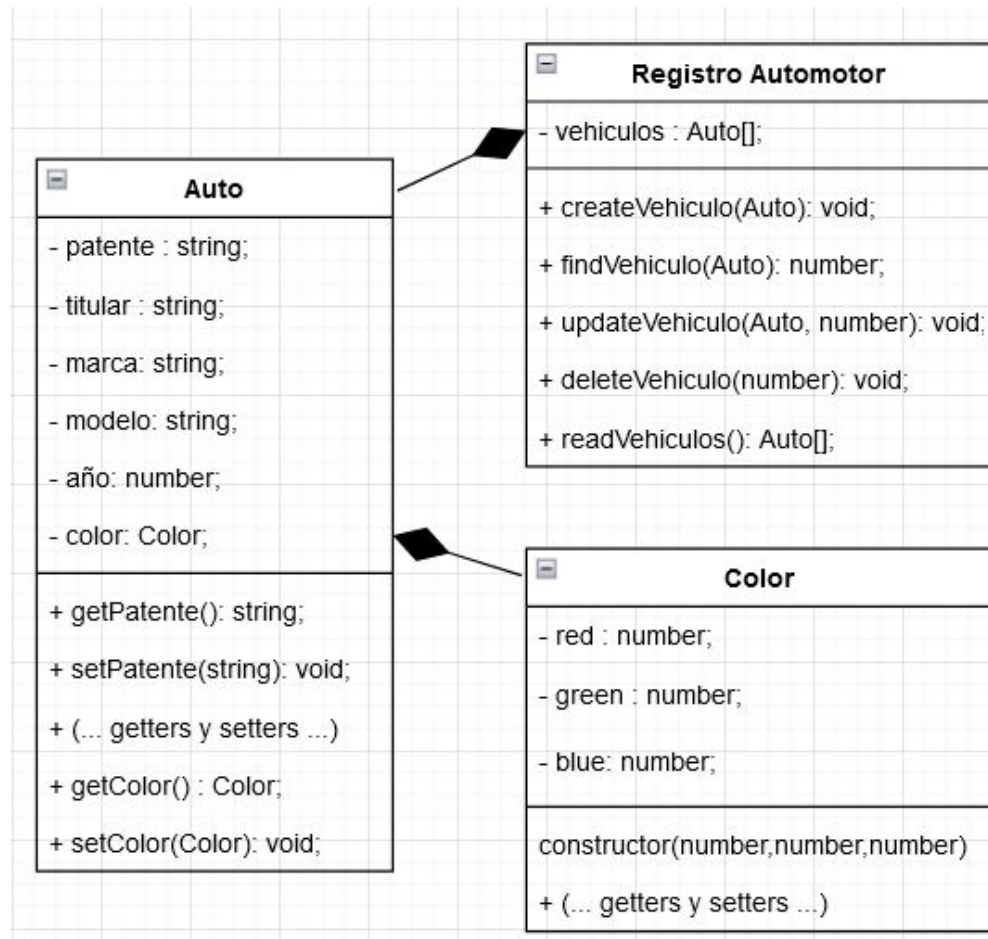
Ejercicio II

Inciso 1

- Implementar las clases del diagrama:

Inciso 2

- Utilizar la librería de manejo de archivos de texto para guardar y recuperar la información creada en el uso de la clase RegistroAutomotor



Ejercicio III

Inciso 1

- Realizar el diagrama de la clase elegida (el ejemplo de objeto que pensó).

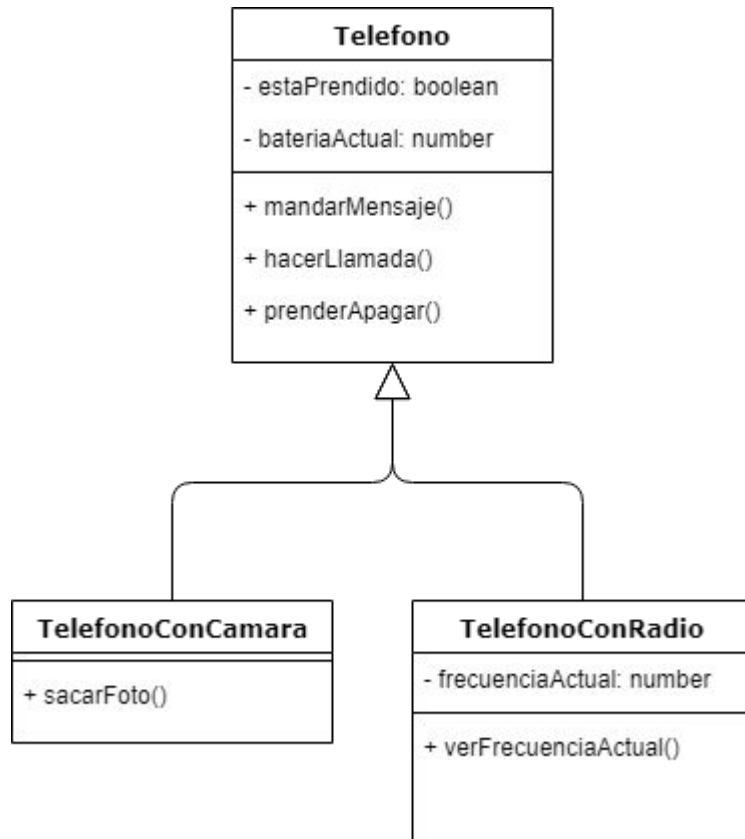
Inciso 2

- Realizar el diagrama de la clase que está compuesta por un arreglo de clases del punto anterior

Inciso 3

- Realizar el diagrama de ambas clases y su relación

Ejercicio IV



Inciso 1

- Implementar las clases y métodos que se muestran

Inciso 2

- Agregar variables/métodos adicionales

Ejercicio V

Inciso 1

- Re-diseñar la solución del RegistroAutomotor, agregando el concepto de herencia para administrar autos, camionetas y camiones, agregando la capacidad del vehículo y la cantidad de ejes.
- Implementar en TypeScript.

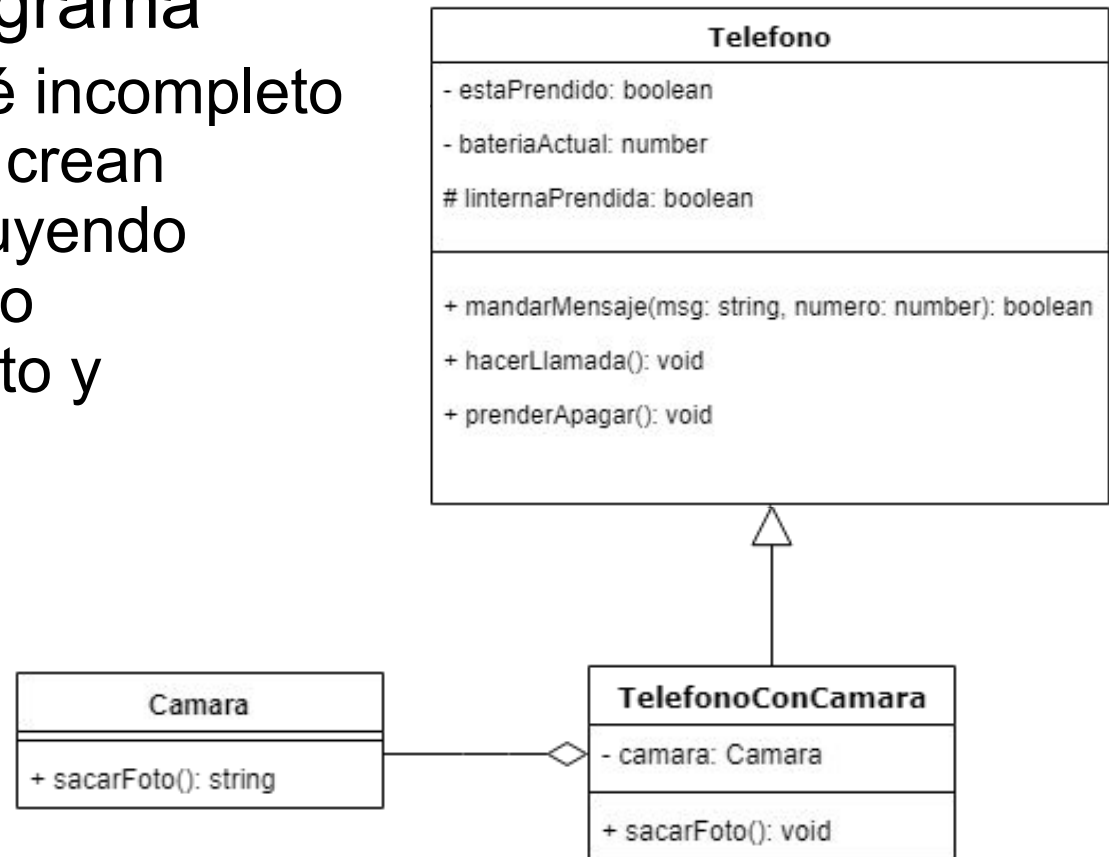
Inciso 2

- Utilizar la librería de manejo de archivos de texto para guardar y recuperar la información creada en el uso de la nueva versión de la clase RegistroAutomotor

Ejercicio VI

Implementar el diagrama

- Puede que esté incompleto
- Agregar lo que crean necesario, incluyendo conceptos como encapsulamiento y polimorfismo.



Ejercicio VII

Usar los conceptos y recomendaciones vistas hasta aquí, para armar una base de datos de publicaciones.

- Hacer el planteo de las clases necesarias
- Implementar diferenciando las publicaciones en: Libro, Diario y Revista.
- Implementar la clase GestorPublicaciones → debe soportar insertar/consultar/modificar/eliminar publicaciones (la entrada de información por teclado) -
CONTEXTO: Pensarlo como una biblioteca pública, en la cual los usuarios pueden consultar estos tipos de publicaciones.
- Luego incorporar en donde se crea necesario un mecanismo para leer publicaciones desde un archivo de texto