Implementação

O objetivo era integrar uma biblioteca antiga, que trabalha com strings, com uma nova abordagem baseada em objetos. Como o método legado (storeVehicleData) só aceita strings e a nova interface espera um objeto Vehicle, usamos o padrão Adapter para fazer essa ponte.

Criei uma classe VehicleStorageAdapter que implementa conceitualmente a interface IVehicleStorage. Como estamos em JavaScript, a interface foi simulada apenas para deixar claro nosso contrato de métodos.

Dentro do adapter, instanciei a classe legada OldVehicleStorage e, no método saveVehicleData, fiz a conversão dos dados do objeto Vehicle para uma string no formato esperado, como: ID: 123, Model: Volvo FH, Year: 2020.

O main.js demonstra como, graças ao adapter, podemos continuar usando a biblioteca antiga sem alterar seu código, mas agora de forma compatível com o novo padrão da empresa.

Desafios e soluções

1. **Ausência de Interfaces em JavaScript**:
   JavaScript não possui suporte nativo a interfaces. Para resolver isso, criei uma classe base IVehicleStorage com um método abstrato que lança erro. Isso ajuda a lembrar que o método saveVehicleData deve ser implementado.
2. **Conversão de dados de objeto para string**:
   Como o método da biblioteca antiga espera uma string bem específica, foi importante garantir que todos os campos relevantes de Vehicle fossem incluídos no formato certo.