

Documento Explicativo - Implementação do Padrão Adapter

Este documento descreve a implementação do padrão de projeto Adapter no contexto de um sistema de gerenciamento de dados veiculares para uma empresa de logística.

Objetivo:

O objetivo foi permitir que a biblioteca legada (OldVehicleStorage), que armazena dados de veículos como String, fosse compatível com a nova interface (IVehicleStorage), que utiliza objetos da classe Vehicle.

Estrutura das Classes:

- Vehicle: classe simples representando um veículo com id, modelo e ano.
- IVehicleStorage: interface moderna que define o método saveVehicleData(Vehicle).
- OldVehicleStorage: biblioteca legada com o método storeVehicleData(String).
- VehicleStorageAdapter: implementa IVehicleStorage e adapta a chamada ao método da biblioteca legada.

Funcionamento do Adapter:

A classe VehicleStorageAdapter recebe uma instância de OldVehicleStorage. No método saveVehicleData(Vehicle vehicle), os dados do veículo são extraídos e formatados em uma String legível, que é então repassada ao método storeVehicleData da biblioteca legada.

Exemplo de conversão:

```
Vehicle(id='123ABC', model='Caminhão Volvo', year=2020)
```

```
=> 'ID: 123ABC, Modelo: Caminhão Volvo, Ano: 2020'
```

Classe Main:

Na classe de teste Main, criamos um objeto Vehicle e o armazenamos utilizando a interface moderna IVehicleStorage, demonstrando a total abstração do uso da biblioteca legada.

Desafios Encontrados:

O principal desafio foi realizar a conversão de um objeto orientado a objetos para uma estrutura legada baseada em String, mantendo clareza e desacoplamento. A solução foi centralizar essa

lógica de conversão dentro do Adapter, mantendo o restante do sistema coeso e aderente ao SOLID.

Conclusão:

O padrão Adapter foi aplicado com sucesso para integrar sistemas com interfaces incompatíveis. A solução promove reutilização de código legado e facilita a evolução do sistema com um baixo custo de manutenção.