Implementação do Padrão de Projeto Adapter em Java

Aluna: Nathany Eleutério
Turma: 5° Período – Padrões de Projeto

Objetivo

Esta atividade teve como propósito aplicar o padrão de projeto Adapter em uma situação realista, simulando a integração entre uma biblioteca legada e uma interface moderna de armazenamento de dados de veículos. A proposta visava demonstrar a capacidade do padrão Adapter em promover compatibilidade entre sistemas que não foram originalmente projetados para trabalhar juntos, sem a necessidade de alterar o código legado.

Estrutura da Solução

O sistema foi desenvolvido em Java com as seguintes classes:

Vehicle: Representa o modelo de dados do veículo (com id, model e year).

OldVehicleStorage: Representa a biblioteca legada, que armazena os dados em formato de String via o método storeVehicleData(String data).

IVehicleStorage: Interface moderna, que define o método saveVehicleData(Vehicle vehicle) utilizando um objeto fortemente tipado.

VehicleStorageAdapter: Classe que implementa a interface moderna e adapta os dados do objeto Vehicle para o formato esperado pela biblioteca legada.

Main: Classe principal de teste, onde é criada a instância do veículo e a integração é demonstrada com sucesso usando o Adapter.

Funcionamento do Adapter

A classe VehicleStorageAdapter atua como uma ponte entre o novo e o antigo sistema. Ela implementa a interface IVehicleStorage e internamente utiliza uma instância de OldVehicleStorage. Ao receber um objeto Vehicle, ela converte suas informações em uma String no formato:

```
ID=xxx|MODEL=xxx|YEAR=xxxx
```

Essa string é então passada para o método storeVehicleData() da biblioteca legada.

```
String legacyData = String.format("ID=%s|MODEL=%s|YEAR=%d",
    v.getId(), v.getModel(), v.getYear());
```

Dessa forma, conseguimos manter o uso da interface moderna no sistema, sem perder a funcionalidade já existente da biblioteca antiga.

✅ Benefícios Observados
Reaproveitamento de código legado sem precisar reescrevê-lo.

Código modular, limpo e desacoplado, facilitando futuras manutenções.

Aplicação prática de um padrão de projeto do tipo estrutural, com separação clara de responsabilidades.

Facilidade para migrar futuramente para um sistema 100% moderno, bastando apenas trocar a implementação da interface IVehicleStorage.

⚠️ Desafios Enfrentados
Garantir a conversão correta dos dados entre os dois formatos (objeto e string).

Manter a simplicidade e clareza do código, mesmo com a introdução de um padrão adicional.

Simular um ambiente realista de biblioteca legada sem criar complexidade desnecessária.

📌 Conclusão
A atividade demonstrou com clareza como o padrão Adapter pode ser utilizado para resolver incompatibilidades entre sistemas legados e modernos. O código foi escrito de forma organizada, com separação de responsabilidades e seguindo os princípios do design orientado a objetos.

Esse padrão é especialmente útil em projetos em transição tecnológica, onde há a necessidade de manter sistemas antigos funcionando enquanto novas soluções são desenvolvidas.