Resumo TEMA 4 **Paulo Henrique 202302294308**
Padrão Comportamental GoF com exemplo

---

1. Chain of Responsibility

- **Intenção:** Evita acoplamento entre remetente e receptor da requisição. Encadeia objetos e passa a requisição até que um deles a processe.
- **Exemplo:** Aprovação de desconto: vendedor → supervisor → gerente.

---

2. Command

- **Intenção:** Encapsula uma requisição como objeto, permitindo parametrizar clientes, enfileirar, registrar histórico, desfazer/refazer operações.
- **Exemplo:** Jogo em que cada tecla executa uma ação (andar, pular) configurada dinamicamente.

---

3. Interpreter

- **Intenção:** Define gramática e interpreta expressões conforme a gramática.
- **Exemplo:** Avaliação de expressões matemáticas: (a + b) * c.

---

4. Iterator

- **Intenção:** Fornece uma maneira de acessar elementos de um agregado sequencialmente, sem expor sua estrutura interna.
- **Exemplo:** Percorrer produtos em uma ArrayList sem conhecer a implementação.

---

5. Mediator

- **Intenção:** Encapsula a interação entre um conjunto de objetos, promovendo o desacoplamento.
- **Exemplo:** Em um sistema de e-commerce, o mediador coordena pagamento, estoque, envio e notificação de email.

---

6. Memento

- **Intenção:** Captura e externa o estado interno de um objeto sem violar seu encapsulamento, permitindo restaurar esse estado depois.
- **Exemplo:** "Desfazer" em um editor de texto — restaurando estados anteriores.

---

7. Observer

- **Intenção:** Define dependência um-para-muitos entre objetos, de modo que quando um objeto muda de estado, todos os seus dependentes são notificados automaticamente.
- **Exemplo:** Atualização automática de gráficos e tabelas em uma planilha ao alterar um dado.

---

8. State

- **Intenção:** Permite que um objeto altere seu comportamento quando seu estado interno muda.
- **Exemplo:** Impressora que muda de comportamento conforme o estado: pronta, imprimindo ou com erro.

---

9. Strategy

- **Intenção:** Define uma família de algoritmos, encapsula cada um e os torna intercambiáveis.
- **Exemplo:** Cálculo de juros utilizando diferentes algoritmos: Regressão Linear, Bootstrap ou Splines.

---

10. Template Method

- **Intenção:** Define o esqueleto de um algoritmo, delegando a subclasses a implementação de alguns passos.
- **Exemplo:** Classe Relatório com método gerar () que chama prepararDados(), formatar (), e imprimir (), sendo que cada relatório implementa esses passos.

---

11. Visitor

- **Intenção:** Permite adicionar novas operações a uma estrutura de objetos sem modificar as classes desses objetos.
- **Exemplo:** Conversor de documentos: a mesma estrutura de objetos pode gerar HTML, PDF ou Base64, conforme o visitante.

Quadro Comparativo dos Padrões Comportamentais GoF

Padrão	Problema Resolvido	Consequência/Benefício	Exemplo
Chain of Responsibility	Desacoplar remetente e vários receptores.	Flexibilidade no processamento de requisições.	Aprovação de desconto em loja.
Command	Encapsular requisições como objetos.	Histórico, desfazer/refazer, enfileirar comandos.	Controle de ações em jogos.
Interpreter	Avaliar e interpretar gramáticas/expressões.	Implementação de DSLs (Domain Specific Language).	Avaliação de expressões matemáticas.
Iterator	Percorrer coleções sem expor detalhes internos.	Uniformiza e desacopla o acesso às coleções.	Percorrer uma ArrayList em Java.
Mediator	Simplificar comunicação complexa entre muitos objetos.	Reduz acoplamento e facilita manutenção.	Processo de compra em e-commerce.
Memento	Salvar e restaurar estado de objetos.	Implementação de "desfazer" (undo).	Undo/Redo em editores de texto.
Observer	Notificar automaticamente múltiplos objetos dependentes.	Sincronização automática entre dados e visualizações.	Atualização de gráficos em tempo real.
State	Mudar o comportamento de um objeto conforme seu estado.	Evita múltiplos if/else ou switch.	Impressora com estados: pronta, erro etc.
Strategy	Definir e trocar algoritmos em tempo de execução.	Flexibiliza escolha de algoritmos sem alterar clientes.	Cálculo de juros com diversos métodos.
Template Method	Definir algoritmo padrão, mas permitir alteração de etapas específicas.	Reutilização de código base e especialização controlada.	Geração de relatórios.
Visitor	Adicionar operações sem alterar classes dos elementos.	Mantém classes estáveis, permite novas funcionalidades.	Conversão de dados: HTML, PDF, Base64 etc.