

ATIVIDADE::Detecção de Code Smells

Atividades 1 & 2: Detecção e Melhoria de Code Smells (Exemplo da Locadora)

1. Identificação de Code Smells no Projeto (Versão Inicial):

Analisando o código inicial da locadora (especificamente o método `conta()`), que é o foco principal da refatoração), podemos identificar vários *code smells* definidos no próprio documento:

- **Método Longo (Inflador):** O método `conta()` é o exemplo clássico de um "Método Longo". Ele acumula muitas responsabilidades: calcula o valor de cada locação, soma os pontos de locador frequente, calcula o total e formata a string de resultado. A própria aula o descreve como "Um método longo como esse".
- **Comandos Switch (Abusador da OO):** O método `conta()` contém um switch complexo para determinar o preço com base no `códigoPreço` do filme. Isso é um *code smell* do tipo "Comandos Switch", que geralmente indica uma falha em usar polimorfismo.
- **Inveja dos Dados (Acoplador):** Quando o cálculo do preço é extraído para o método `quantiaDe` na classe `Cliente`, esse novo método exibe "Inveja dos Dados". Ele acessa muito mais dados de *outro* objeto (no caso, `umaLocação` e, por extensão, `Filme`) do que de sua própria classe (`Cliente`).
- **Código Duplicado (Dispensável):** O documento levanta a hipótese de criar um `contaHtml()`. Se o programador segue o caminho do "copiar, colar e adaptar o código", ele criaria "Código Duplicado", um *smell* da categoria "Dispensáveis". A lógica de cálculo de preço e pontos seria replicada em dois métodos.

2. Como Melhorar o Projeto e Remover os Code Smells:

O próprio documento demonstra as refatorações necessárias, que servem como tratamento para os *smells* identificados:

- Para "Método Longo" (`conta()`): A refatoração aplicada é **Extract Method**.
 - Solução: O bloco switch é extraído do `conta()` para um novo método, `quantidade(cada)`, que depois é renomeado para `quantiaDe(umaLocação)`.
- Para "Inveja dos Dados" (`quantiaDe()`): A refatoração é **Move Method**.
 - Solução: O método `quantiaDe`, que estava em `Cliente`, é movido para onde os dados estão: primeiro para a classe `Locação` (e renomeado `lerPreço`) e, mais tarde, para a classe `Filme`.
- Para "Código Duplicado" (potencial): A refatoração é **Replace Temp with Query** (Substituir Variáveis Temporárias por Consultas).

ATIVIDADE::Detecção de Code Smells

- Solução: As variáveis temporárias que calculavam os totais (quantiaTotal e pontosLocadorFrequente) são substituídas por métodos de consulta (lerPreçoTotal() e lerTotalPontosLocadorFrequente()). Isso permite que tanto o conta() quanto o novo contaHtml() reutilizem a mesma lógica de cálculo, evitando duplicação.
- Para "Comandos Switch": A refatoração é Replace Conditional with Polymorphism (ou, no caso, "Replace Type Code with State/Strategy").
 - Solução: Esta é a mudança mais profunda. O códigoPreço é substituído por um objeto Preço que segue o padrão State. A lógica do switch é distribuída entre classes polimórficas (PreçoNormal, PreçoInfantil, PreçoLançamento).

Atividade 5: Ferramentas de Detecção de Code Smells (JAVA)

Abaixo está uma pesquisa e uma tabela comparativa de três ferramentas populares para detecção de *code smells* em código JAVA.

Ferramenta	Foco Principal	Exemplo de <i>Code Smells</i> Detectados	Integração
SonarQube (com SonarJava)	Qualidade e Segurança Contínua	É uma plataforma que agrega <i>Bugs</i> , <i>Vulnerabilidades</i> e <i>Code Smells</i> . Cobre uma vasta gama, incluindo "Método Longo", "Complexidade Ciclomática" alta, "Código Duplicado" e vulnerabilidades de segurança.	Servidor web independente, plugins para IDEs (SonarLint), e integração com CI/CD (Maven, Gradle).
PMD	Más Práticas de Programação	Foca em "más práticas", "código morto" (unused variables), <i>catch blocks</i> vazios, "complexidade excessiva" e uso ineficiente de objetos.	Plugins para a maioria das IDEs (Eclipse, IntelliJ) e ferramentas de <i>build</i> (Maven, Ant).
Checkstyle	Padrões de Estilo de Código	Foca primariamente em "convenções" e "estilo", como formatação (uso de chaves, espaçamento), convenções de nomenclatura e "magic numbers". Embora focado em estilo, o não cumprimento de padrões é considerado um <i>code smell</i> por muitos.	Plugins para IDEs e ferramentas de <i>build</i> . Muitas vezes usado em conjunto com PMD.