

fix the longmethod code smell using the extract method refactoring technique

8 GitHub Copilot

Vou ajudar você a refatorar o código para corrigir o code smell "Long Method" no método getReferenceCountMap() usando a técnica Extract Method.

Passo a passo:

- 1. Dividir o método getReferenceCountMap() em métodos menores e mais específicos
- 2. Criar métodos separados para contar cada tipo de referência
- 3. Manter o método original mais limpo e legível

O arquivo precisa ser modificado para extrair a lógica de contagem de cada tipo de referência em métodos separados.

■ StudyMaterial.java

Principais mudanças:

- 1. Extraí a inicialização do mapa para o método initializeCountMap()
- 2. Criei um método countAllReferences() para iterar sobre as referências
- 3. Extraí a lógica de contagem específica para cada tipo em métodos separados
- 4. Substituí o uso de getClass() por instanceof que é mais seguro e legível
- 5. Os novos métodos são privados pois são detalhes de implementação internos







você adicionou smells de feature envy nos métodos countVideoReference e countTextReference

& GitHub Copilot

Você está certo! Vamos corrigir o Feature Envy movendo a lógica de contagem para suas respectivas classes, seguindo o princípio Tell Don't Ask.

Passos para corrigir:

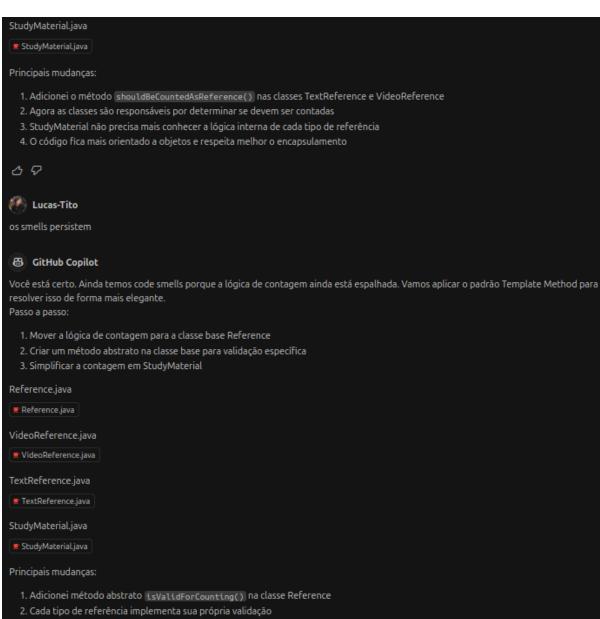
- 1. Mover a lógica de contagem para as classes VideoReference e TextReference
- 2. Simplificar os métodos de contagem em StudyMaterial

TextReference.java

■ TextReference.java

VideoReference.java

■ VideoReference.java



- 3. Adicionei método getReferenceTypeNane() para padronizar os nomes das chaves
- 4. Simplifiquei a lógica de contagem em um único método
- 5. Removi os métodos específicos de contagem que estavam duplicando código

Esta solução:

- Remove a duplicação de código
- Centraliza a lógica de contagem
- Torna mais fácil adicionar novos tipos de referência
- Segue o princípio Open/Closed
- Resolve o Feature Envy





Lucas-Tito

o seguinte erro está sendo retornado:

java: org.example.studymaterial.AudioReference is not abstract and does not override abstract method isValidForCounting() in org.example.studymaterial.Reference

GitHub Copilot

Ah sim! Precisamos implementar o método abstrato isValidForCounting() na classe AudioReference também, já que ela herda de

AudioReference.java

AudioReference.java

Explicação:

- · Para AudioReference, como não há validações específicas mencionadas, vamos assumir que todos os áudios são válidos para contagem
- Se no futuro houver necessidade de validações específicas para áudio, podemos modificar esta implementação

