



fix the code smell The method `getReferenceCountMap()` has a NCSS line count of 21. using "Extract method" refactoring technique...



Java



```
public Map<String, Integer> getReferenceCountMap() {
    Map<String, Integer> response = new HashMap<>();
    response.put("Audio References", 0);
    response.put("Video References", 0);
    response.put("Text References", 0);

    countReferences(response);

    setReferenceCount(response);
    return response;
}

private void countReferences(Map<String, Integer> response) {
    for (Reference reference : references) {
        if (reference.getClass() == AudioReference.class) {
            incrementCount(response, "Audio References");
        } else if (reference.getClass() == VideoReference.class &&
            ((VideoReference) reference).handleStreamAvailability()) {
            incrementCount(response, "Video References");
        } else if (reference.getClass() == TextReference.class &&
            ((TextReference) reference).handleTextAccess()) {
            incrementCount(response, "Text References");
        }
    }
}

private void incrementCount(Map<String, Integer> response, String key) {
    Integer count = response.get(key);
    response.put(key, count + 1);
}
```

Explanation of Changes:

Extracted Method: The core logic of counting references within the loop is extracted into a separate private method `countReferences(Map<String, Integer> response)`. This improves readability and reduces the original method's size.

Simplified Conditional Checks: The `if-else` conditions are slightly simplified by combining the class check and the specific condition within the `if` statement itself.



Introduced Helper Method: A new private method `incrementCount(Map<String, Integer> response, String key)` is introduced to handle the common logic of incrementing the count for a given key in the map. This further improves code reusability and reduces redundancy.

These changes effectively address the "NCSS line count of 21" code smell by breaking down the original method into smaller, more focused units. This enhances code maintainability, readability, and testability.



Powered by Gemini All Chat Downloader