



fix the code smell "The method kanbanView() has a NCSS line count of 31." using "Extract method" refactoring technique

...



Java



```
public String kanbanView() throws Exception {
    try{
        if(kanban.isEmpty()){
            throw new Exception("No material found");
        }

        StringBuilder sb = new StringBuilder();
        sb.append("[ ");
        sb.append(getMaterialStateAsString(State.TODO, "Material ToDo:"));
        sb.append(System.lineSeparator());
        sb.append(getMaterialStateAsString(State.DOING, "Material in progress:"));
        sb.append(System.lineSeparator());
        sb.append(getMaterialStateAsString(State.DONE, "Material completed:"));
        sb.append(System.lineSeparator());
        sb.append("]");
        return sb.toString();
    } catch (Exception e){
        throw new Exception(e.getMessage());
    }
}

private String getMaterialStateAsString(State state, String stateName) {
    StringBuilder sb = new StringBuilder();
    sb.append(stateName);
    sb.append(System.lineSeparator());

    if(kanban.get(state).isEmpty()){
        sb.append("No material found");
    } else {
        for(PlannerMaterial material : kanban.get(state)){
            sb.append(", ").append(material.toString());
        }
    }
    return sb.toString();
}
```

Explanation:

Extracted Method: The code for handling the logic for each state (TODO, DOING, DONE) has been extracted into a separate private method `getMaterialStateAsString`. This method takes the `state` and the corresponding `stateName` as parameters.



Improved Readability: By extracting the repetitive logic into a separate method, the `kanbanView()` method becomes significantly more concise and easier to read. The extracted method also has a clear and descriptive name, making the code more self-documenting.

Reduced Cyclomatic Complexity: Extracting the state handling logic reduces the cyclomatic complexity of the `kanbanView()` method, making it easier to understand, maintain, and test.

Improved Reusability: The `getMaterialStateAsString` method can now be reused if similar logic is needed elsewhere in the code.

This refactoring addresses the code smell by reducing the number of lines in the `kanbanView()` method, improving its readability, and making it more maintainable.

it does not solve. i have this smell now: "The method `kanbanView()` has a NCSS line count of 15." you are almost there



You're right, the refactoring I provided still has a relatively high NCSS for `kanbanView()`. Let's refine it further to address this.

Java




```
public String kanbanView() throws Exception {
    try{
        if(kanban.isEmpty()){
            throw new Exception("No material found");
        }

        StringBuilder sb = new StringBuilder();
        sb.append("[");
        sb.append(getKanbanStatesAsString());
        sb.append("]");
        return sb.toString();
    } catch (Exception e){
        throw new Exception(e.getMessage());
    }
}

private String getKanbanStatesAsString() {
    StringBuilder sb = new StringBuilder();
    for (State state : State.values()) {
        sb.append(getMaterialStateAsString(state));
        sb.append(System.lineSeparator());
    }
    return sb.toString();
}

private String getMaterialStateAsString(State state) {
    StringBuilder sb = new StringBuilder();
    sb.append(state.name()).append(":").append(System.lineSeparator());
}
```

 1.5 Flash

```
if(kanban.get(state).isEmpty()){
    sb.append("No material found");
} else {
    for(PlannerMaterial material : kanban.get(state)){
        sb.append(", ").append(material.toString());
    }
}
return sb.toString();
}
```

Testar o Gemini Advanced

Explanation:

Further Extraction: The logic for iterating over the `State` enum and building the string for each state has been extracted into a separate method `getKanbanStatesAsString`.

Reduced `kanbanView()`: The `kanbanView()` method now focuses solely on the core logic of constructing the final string, delegating the state-specific logic to the `getKanbanStatesAsString` method.

Improved Maintainability: This further refactoring improves the maintainability of the code by further breaking down the logic into smaller, more focused methods.

This revised version should significantly reduce the NCSS of the `kanbanView()` method and improve its overall readability and maintainability.



Powered by Gemini All Chat Downloader