



fix the code smell The method setGoalSummary() has a NCSS line count of 19." using "Extract method" refactoring technique

...



Java



```
public String setGoalSummary(){
    StringBuilder summary = new StringBuilder();
    summary.append("Goal Summary:\n").append("\n\n");

    appendActiveGoalIfApplicable(summary);
    appendCompletedGoalIfApplicable(summary);
    appendRequirementsIfApplicable(summary);
    appendPlanIfApplicable(summary);
    appendObjectiveIfApplicable(summary);

    this.summary = summary.toString();
    return summary.toString();
}

private void appendActiveGoalIfApplicable(StringBuilder summary) {
    if(this.isActive){
        summary.append("Active Goal:\n").append(goal).append("\n\n");
    }
}

private void appendCompletedGoalIfApplicable(StringBuilder summary) {
    if(this.isCompleted){
        summary.append("Completed Goal:\n").append(goal).append("\n\n");
    }
}

private void appendRequirementsIfApplicable(StringBuilder summary) {
    if(this.goalRequirements != null){
        summary.append("Requirements:\n");
        for(String requirement : this.goalRequirements){
            summary.append(requirement).append(", ");
        }
    }
}

private void appendPlanIfApplicable(StringBuilder summary) {
    if(this.studyPlan != null){
        summary.append("Plan:\n");
        summary.append(this.studyPlan.toString());
    }
}
```

1.5 Flash

```
private void appendObjectiveIfApplicable(StringBuilder summary) {  
    if(this.studyObjective != null){  
        summary.append("Objective:\n");  
        summary.append(this.studyObjective.toString());  
    }  
}
```

Explanation:

- **Extracted Methods:** The original `setGoalSummary()` method has been broken down into smaller, more focused methods:
 - `appendActiveGoalIfApplicable()`
 - `appendCompletedGoalIfApplicable()`
 - `appendRequirementsIfApplicable()`
 - `appendPlanIfApplicable()`
 - `appendObjectiveIfApplicable()`
- **Improved Readability:**
 - The main `setGoalSummary()` method now has a clearer flow, focusing on the overall structure of the summary.
 - The extracted methods have descriptive names, making the code easier to understand and maintain.
- **Reduced Cyclomatic Complexity:** By extracting the conditional logic into separate methods, the cyclomatic complexity of the original method is reduced.
- **Improved Testability:** Each extracted method can be tested independently, making unit testing more efficient and reliable.

This refactoring addresses the "NCSS line count of 19" issue by breaking down the large method into smaller, more manageable units. This improves code readability, maintainability, and testability.



Powered by Gemini All Chat Downloader