



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

...



Java



```
public String toString() {
    StringBuilder str = new StringBuilder();
    for (ToDo toDo : toDos) {
        str.append(toDo.toString());
        str.append("\n");
        str.append(getTrackInfo(toDo));
    }
    String response = str.toString();
    if(response.isEmpty()){
        return "No ToDos found";
    }
    return response;
}

private String getTrackInfo(ToDo toDo) {
    StringBuilder str = new StringBuilder();
    Integer id = toDo.getId();
    List<LocalDateTime> todosDate = this.tracker.get(id);
    if(todosDate == null){
        str.append("No tracks found\n");
    }else{
        for (LocalDateTime ldt : todosDate) {
            String pattern = "yyyy-MM-dd HH:mm:ss";
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern);
            String formattedDate = formatter.format(ldt);
            str.append(formattedDate);
            str.append("\n");
        }
    }
    return str.toString();
}
```

Explanation:

Extract Method: We extracted the code that handles the tracking information (retrieving dates from tracker , formatting them, and appending them to the StringBuilder) into a separate private method called getTrackInfo(ToDo toDo) .

Improved Readability: This refactoring significantly improves the readability of the toString() method. The main logic of the toString() method now focuses on iterating through the toDos list and appending

their information to the `StringBuilder`. The details of how to retrieve and format tracking information are encapsulated within the `getTrackInfo()` method.

Testar o Gemini Advanced

1.5 Flash

Reduced Cyclomatic Complexity: By extracting the tracking information logic, we reduce the number of nested blocks within the `toString()` method, thereby lowering its cyclomatic complexity and making it easier to understand and maintain.

This refactored code addresses the "NCSS line count of 21" issue by breaking down the `toString()` method into smaller, more manageable units. This approach enhances code maintainability, testability, and overall code quality.

it does not solve. i get this smell: "The method `getTrackInfo(ToDo)` has a NCSS line count of 14." resolve using "Extract method" refactoring technique



Python



```
class Tracker:
    def __init__(self):
        self.data = {}

    def track(self, todo_id, date_time):
        if todo_id not in self.data:
            self.data[todo_id] = []
        self.data[todo_id].append(date_time)

    def get_track_info(self, todo_id):
        dates = self.data.get(todo_id)
        if dates is None:
            return "No tracks found\n"
        return format_track_dates(dates)

    def format_track_dates(dates):
        return "\n".join([format_date_time(date) for date in dates]) + "\n"

    def format_date_time(date_time):
        return date_time.strftime("%Y-%m-%d %H:%M:%S")
```

Explanation:

Extract Method: We extracted the code that formats the dates into a separate function called `format_track_dates(dates)`.

Improved Readability: This refactoring improves the readability of the `get_track_info()` method by separating the core logic (checking for dates and returning "No tracks found") from the date formatting details.

Reduced Cyclomatic Complexity: By extracting the date formatting logic, we reduce the number of lines and potential complexity within the `get_track_info()` method. Testar o Gemini Advanced



1.5 Flash

This refactored code addresses the "NCSS line count of 14" issue in `getTrackInfo()` by breaking down the method into smaller, more manageable units, enhancing code maintainability and readability.

its not in python, is the java code that you provide me before.
this one:

...



Java



```
public String toString() {
    StringBuilder str = new StringBuilder();
    for (ToDo toDo : todos) {
        str.append(toDo.toString());
        str.append("\n");
        str.append(getTrackInfo(toDo));
    }
    String response = str.toString();
    if(response.isEmpty()){
        return "No ToDos found";
    }
    return response;
}

private String getTrackInfo(ToDo toDo) {
    Integer id = toDo.getId();
    List<LocalDateTime> todosDate = this.tracker.get(id);
    if(todosDate == null){
        return "No tracks found\n";
    }
    return formatTrackDates(todosDate);
}

private String formatTrackDates(List<LocalDateTime> todosDate) {
    StringBuilder str = new StringBuilder();
    for (LocalDateTime ldt : todosDate) {
        String pattern = "yyyy-MM-dd HH:mm:ss";
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern);
        String formattedDate = formatter.format(ldt);
        str.append(formattedDate);
        str.append("\n");
    }
    return str.toString();
}
```

Explanation:



Extract Method: We extracted the code that iterates through the `todosDate` list, formats each `LocalDateTime`, and appends it to the `StringBuilder` into a separate private method called `formatTrackDates(List<LocalDateTime> todosDate)`.

Improved Readability: This refactoring improves the readability of the `getTrackInfo()` method by separating the core logic (checking for the existence of `todosDate` and returning "No tracks found" if it's null) from the date formatting and appending logic.

Reduced Cyclomatic Complexity: By extracting the date formatting and appending logic, we reduce the number of lines and potential complexity within the `getTrackInfo()` method.

This refactored code addresses the "NCSS line count of 14" issue in `getTrackInfo()` by breaking down the method into smaller, more manageable units, enhancing code maintainability and readability.



Powered by Gemini All Chat Downloader