

Using AI to Refactor Legacy Code:

A Practical Guide

Scott Wierschem

<https://KeepCalmAndRefactor.com>

Original Java Code

```
import java.util.*;

public static List<String>
stringsOfLength5InUpperCase(List<String> strings) {
    List<String> result = new ArrayList<>();
    strings.stream()
        .map(String::toUpperCase)
        .filter(string -> string.length() == 5)
        .forEach(result::add);
    return result;
}
```

Differences & Unique Contributions by Tool



- **ChatGPT:**

- Emphasized misuse of `forEach` in streams
- Warned about possible `.toUpperCase()` Unicode effects
- Recommended import cleanup and discussed code semantics



- **MS Copilot:**

- Warned `.toUpperCase()` may change string length (Unicode concern)
- Suggested filtering before mapping



- **Google Gemini:**

- Emphasized idiomatic stream usage
- Discussed immutability and stream clarity
- Did not mention the `toUpperCase()` length issue

Differences & Unique Contributions by Tool



- **Grok:**



- Identified thread-safety risks with `forEach` and `ArrayList` in parallel streams
- Recommended null-checks
- Noted output order preservation and provided actual output



- **Claude:**

- Focused on semantic correctness of stream order
- Explicitly recommended filter before map
- Retained `forEach(result::add)`

AI Code Review Comparison

Focus Area	ChatGPT 	Copilot 	Gemini 	Grok 	Claude 
Use of collect() instead of forEach	✓	✗	✓	✓	⚠
Warns about toUpperCase() length impact	✓	✓	✗	✓	✓
Advises filter before map	✓	✓	✗	⚠	✓
Mentions thread safety	⚠	✗	✗	✓	✗
Null input handling	✗	✗	✗	✓	✗
Explains actual output	✗	⚠	✗	✓	✓

Final Recommended Code

```
import java.util.List;
import java.util.stream.Collectors;

public static List<String>
stringsOfLength5InUpperCase(List<String> strings) {
    if (strings == null) {
        return Collections.emptyList();
    }
    return strings.stream()
        .filter(s -> s.length() == 5)
        .map(String::toUpperCase)
        .collect(Collectors.toList());
}
```

Debugging with Junie





The Debugging Scenario

****Problem:****


- Method ``getTableDataItem()`` on ``DataLoaderTableModel`` returns ``null``.
- Goal: Identify why and fix the test setup.

****Approach:****

- Use AI to analyze structure, code, and method behavior step by step.

Prompt: Stating the Problem

> The initialization of the DataLoaderTableModel object is not working as I expected. Calling `getTableDataItem()` on that object always returns a null. How can I properly initialize the data so that the function call returns a value?



> Get structure

- Reviewed structure of `DataLoaderTableModel`
- Focused on why `getTableDataItem()` returns `null`

> Open DataLoaderTableModel.java

- Examined `getTableDataItem()` method implementation
- Reviewed how data is initialized

> Open DataLoaderTableModel.java

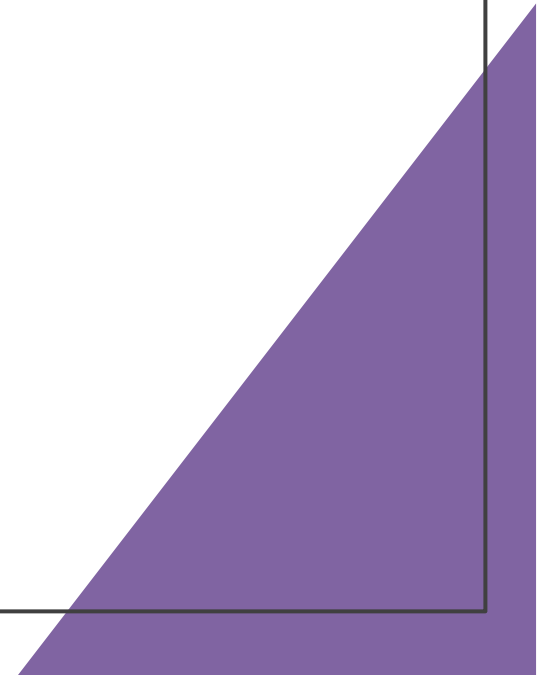
- Reconfirmed the focus on the constructor and data structure initialization

> Get structure

- Analyzed `DataLoaderTest.FakeDataLoaderTableModel`
- Focused on test data initialization

> Open DataLoaderTest.java

- Investigated how `FakeDataLoaderTableModel` is tested
- Verified expected vs. actual data output



> Open DataLoaderDatabaseFunctionsTest.java

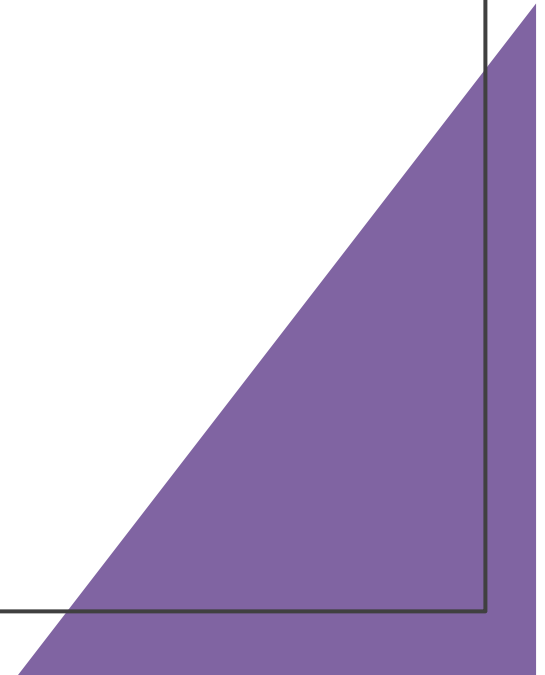
- Continued test data analysis

> Scroll Down

- Confirmed `FakeDataLoaderTableModel` method
`getTableDataItem()` does not show issue in test class

> Search in project "getTableDataItem"


- Found that `FakeDataLoaderTableModel` does ****not**** override `getTableDataItem()`
- Root cause identified: missing override



> The issue is in the `FakeDataLoaderTableModel` class used in your tests.

- Class extends `DataLoaderTableModel`
- Doesn't override `getTableDataItem()`

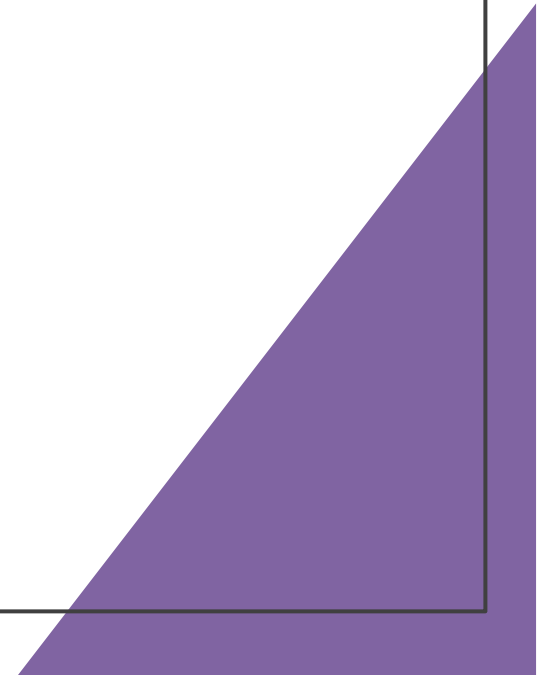
```
@Override
public Object getTableDataItem(int row, int column)
{
    if (row == 0 && column < variableNames.size()) {
        return variableNames.get(column);
    }
    if (column < variableNames.size()) {
        String variableName =
variableNames.get(column);
        return values.get(variableName);
    }
    return null;
}
```



Other tools:



- <https://docs.openrewrite.org/>
- <https://www.qodo.ai/>
- <https://refact.ai/>





Scott Wierschem
@scott_Wierschem
scottwierschem.bsky.social
in/scott-wierschem

KeepCalmAndRefactor.com
scott@KeepCalmAndRefactor.com

<https://KeepCalmAndRefactor.com/code-smells>