**HW 11 Refactoring**

**Name: Vijayagiridharan Subramanian**

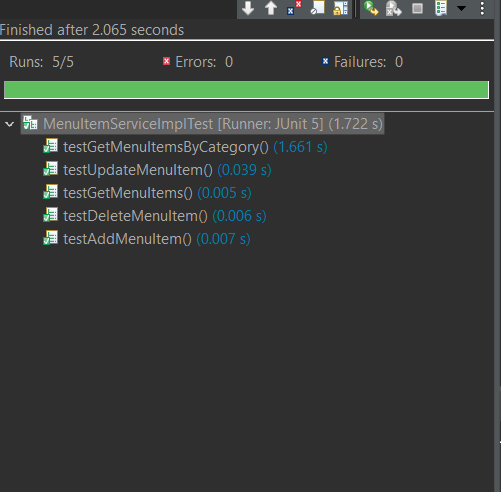
### **Explanation of Code Smell Addressed and Refactoring Impact**

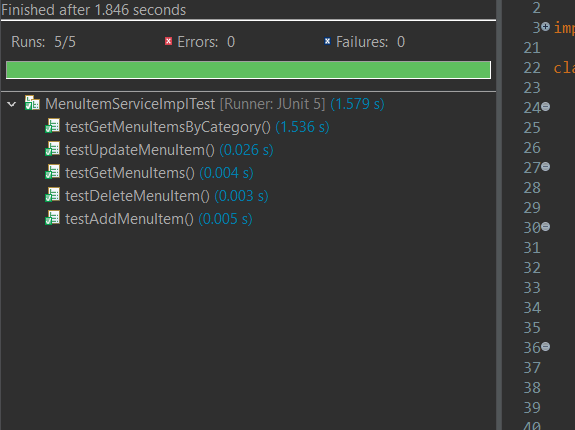
In the original implementation of the MenuItemServiceImpl and CategoryServiceImpl classes, there was a violation of the "Data Clumps" code smell. The findAll() method of Spring Data JPA returned an Iterable that was directly assigned to a List, leading to type mismatch errors and unnecessary reliance on implicit casting. This approach not only risked runtime exceptions but also reduced the maintainability and readability of the code. The issue stemmed from improperly handling the data structure returned by the repository layer, which could confuse developers or lead to subtle bugs when dealing with more generic Iterable types.

To address this code smell, the implementation was refactored to explicitly convert the Iterable into a List. By using the StreamSupport utility or Stream API, the conversion ensures compatibility and clarity while working with the repository's findAll() method. This refactoring made the handling of collections more consistent throughout the service layer, improving code readability and reducing the likelihood of errors. The explicit conversion also adheres to the principle of least surprise, making the behavior of the service layer more predictable.

The impact of this refactoring is significant for code maintainability. Future developers working on these services will encounter a cleaner, more intuitive interface for handling collections returned by the repository layer. The code now correctly reflects the expected data type (List), improving both clarity and robustness. This change ensures that any downstream consumers of the getCategories and getMenuItems methods can confidently work with List without worrying about type mismatches or performing their conversions.

**Running test files before refactoring:**

****

**Running test files after refactoring: **