Java-Removing Items From Collections

Project Abstract

The purpose of this project is to demonstrate how to perform various operations on a collection, specifically an ArrayList, in Java. This task helps you to understand how to use the `ArrayList` class to store elements and perform operations like adding, removing, checking the size, and looping through the collection. The project focuses on using methods such as `add()`, `remove()`, `size()`, and iterating through the collection using an iterator.

This project focuses on:
1. Understanding how to create and use an `ArrayList` in Java.
2. Adding items to a collection using the `add()` method.
3. Removing items from the collection using the `remove()` method.
4. Iterating through the collection using an iterator.
5. Finding the size of the collection using the `size()` method.

Tasks Overview

Task 1: Add, Remove, and Iterate Through List

Objective: Perform various operations on a List (ArrayList), including adding, removing, checking the size, and iterating through the list.

Detailed Description: In this task, you will create an `ArrayList` of strings and perform the following operations: adding elements to the list using `add()`, removing an element using `remove()`, finding the size of the list using `size()`, and looping through the list using an iterator.

Steps:
1. **Create a List (ArrayList):**
   - Declare a `List` object named `list` to store `String` values.

2. **Add Items to the List:**
   - Use the `add()` method to add several string items ("Apple", "Banana", "Cherry") to the `list` and print the list as `"List after adding items: " + list`.

3. **Remove an Item from the List:**
   - Use the `remove()` method to remove an item (remove "Banana").

  - Print the `list` to show the result after removing an item as `"List after removing 'Banana': " + list`.

4. **Loop Through the List:**
   - Use an iterator to loop through the `list` and print each element as `iterator.next()`.

5. **Find the Size of the List:**
   - Use the `size()` method to get and print the size of the `list` as `"Size of List: " + size`.

Execution Steps to Follow:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) □ Terminal □New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To run your project use command:**

   ```
   mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.CollectionOperations"
   ```

7. **To test your project test cases, use the command**

   ```
   mvn test
   ```

8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**