Name of the Course : Ultimate Java Masterclass

Level : Medium

Tool Stack : Java8 and Junit5

Problem Statement : Provide a code solution to accept fruit names in two ArrayLists. Delete fruit names based on conditions from both lists. Combine the both lists and return the final output as a string array. If the array is empty the program will print as “No fruit found”.

Description : Create two classes one FruitList class with two ArrayLists fruitList1 and fruitList2 of FruitList type and a parameterized constructor and another MainClass with two static methods.

1. public static String[] deleteAndMergeFruitLists(int m, ArrayList<String> al, int n, ArrayList<String> al1), which accepts the both ArrayLists with sizes m and n as input. Remove all fruits whose name ends with 'a' or 'e' from first ArrayList and remove all fruits whose name begins with 'm' or 'a' from second ArrayList then combine the two lists and return the final output as a String array. If the array is empty the program will print as “No fruit found”.
2. public static void main method to Read m strings as input (fruit names). Create an arraylist to store the m strings in this ArrayList. Read n strings as input (fruit names). Create an ArrayList to store the n strings in this ArrayList and call the deleteAndMergeFruitLists method to generate string array of fruits.

Code:

**package** yaksha;

**import** java.util.ArrayList;

**public** **class** FruitList {

**private** ArrayList<String> fruitList1;

**private** ArrayList<String> fruitList2;

**public** FruitList(ArrayList<String> fruitList1, ArrayList<String> fruitList2) {

**super**();

**this**.fruitList1 = fruitList1;

**this**.fruitList2 = fruitList2;

}

**public** ArrayList<String> getFruitList1() {

**return** fruitList1;

}

**public** **void** setFruitList1(ArrayList<String> fruitList1) {

**this**.fruitList1 = fruitList1;

}

**public** ArrayList<String> getFruitList2() {

**return** fruitList2;

}

**public** **void** setFruitList2(ArrayList<String> fruitList2) {

**this**.fruitList2 = fruitList2;

}

}

**package** yaksha;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**class** MainClass {

**static** **int** *j* = 0;

**public** **static** String[] deleteAndMergeFruitLists(**int** m, ArrayList<String> al, **int** n, ArrayList<String> al1) {

String a[] = **new** String[m + n];

String mergedList[];

**int** count = 0;

**for** (**int** i = 0; i < m; i++) {

String s1 = al.get(i);

**if** (s1.charAt(s1.length() - 1) != 'a' && s1.charAt(s1.length() - 1) != 'e'

&& s1.charAt(s1.length() - 1) != 'A' && s1.charAt(s1.length() - 1) != 'E') {

a[*j*] = s1;

*j*++;

}

}

**for** (**int** i = 0; i < n; i++) {

String s1 = al1.get(i);

**if** (s1.charAt(0) != 'A' && s1.charAt(0) != 'M' && s1.charAt(0) != 'a' && s1.charAt(0) != 'm') {

a[*j*] = s1;

*j*++;

}

}

**for** (**int** i = 0; i < a.length; i++) {

**if** (a[i] != **null**)

count++;

}

mergedList = **new** String[count];

**for** (**int** i = 0; i < count; i++) {

mergedList[i] = a[i];

}

**return** mergedList;

}

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter size of first ArrayList");

**int** n = sc.nextInt();

ArrayList<String> al = **new** ArrayList<String>();

ArrayList<String> al1 = **new** ArrayList<String>();

System.***out***.println("Enter fruit names");

**for** (**int** i = 0; i < n; i++) {

al.add(sc.next());

}

System.***out***.println("Enter size of second ArrayList");

**int** n1 = sc.nextInt();

System.***out***.println("Enter fruit names");

**for** (**int** i = 0; i < n1; i++) {

al1.add(sc.next());

}

FruitList fruitList = **new** FruitList(al, al1);

String[] ans = MainClass.*deleteAndMergeFruitLists*(n, fruitList.getFruitList1(), n1, fruitList.getFruitList2());

**if** (ans != **null**) {

System.***out***.println("Merged fruitlist is");

**for** (**int** i = 0; i < ans.length; i++)

System.***out***.println(ans[i]);

} **else**

System.***out***.println("No fruit found");

sc.close();

}

}

Pom.xml

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>iiht.yaksha.mqtwo</groupId>

<artifactId>JavaMasterClassMqTwo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>JavaMasterClassMqTwo</name>

<description>JavaMasterClassMqTwo</description>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>${maven.compiler.source}</maven.compiler.target>

<junit.jupiter.version>5.5.2</junit.jupiter.version>

<junit.platform.version>1.5.2</junit.platform.version>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.12</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>${junit.jupiter.version}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.platform</groupId>

<artifactId>junit-platform-runner</artifactId>

<version>${junit.platform.version}</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.2</version>

</plugin>

</plugins>

</build>

</project>

Junit Testing

**package** yaksha;

**import** java.io.File;

**import** java.io.FileWriter;

**import** java.io.IOException;

// boiler-plate code

**public** **class** TestUtils {

**public** **static** File *businessTestFile*;

**public** **static** File *boundaryTestFile*;

**public** **static** File *exceptionTestFile*;

**static** {

*businessTestFile* = **new** File("./output\_revised.txt");

*businessTestFile*.delete();

*boundaryTestFile* = **new** File("./output\_boundary\_revised.txt");

*boundaryTestFile*.delete();

*exceptionTestFile* = **new** File("./output\_exception\_revised.txt");

*exceptionTestFile*.delete();

}

**public** **static** **void** yakshaAssert(String testName, Object result, File file) **throws** IOException {

System.***out***.println("\n" + testName + "=" + result);

FileWriter writer = **new** FileWriter(file, **true**);

writer.append("\n" + testName + "=" + result);

writer.flush();

writer.close();

}

**public** **static** String currentTest() {

**return** Thread.*currentThread*().getStackTrace()[2].getMethodName();

}

}

**package** yaksha;

**import** **static** yaksha.TestUtils.*businessTestFile*;

**import** **static** yaksha.TestUtils.*currentTest*;

**import** **static** yaksha.TestUtils.*yakshaAssert*;

**import** java.util.\*;

**import** org.junit.jupiter.api.Test;

**class** MainClassTest {

@Test

**void** testDeleteAndMergeFruitLists() **throws** Exception {

// Test will pass

ArrayList<String> fruitList1 = **new** ArrayList<String>();

ArrayList<String> fruitList2 = **new** ArrayList<String>();

fruitList1.add("Apple");

fruitList1.add("Cherry");

fruitList1.add("Grapes");

fruitList2.add("Orange");

fruitList2.add("Mango");

fruitList2.add("Melon");

fruitList2.add("Apple");

String expectedResult[] = { "Cherry", "Grapes", "Orange" };

List<String> list1 = Arrays.*asList*(expectedResult);

FruitList fruitList = **new** FruitList(fruitList1, fruitList2);

String result3[] = MainClass.*deleteAndMergeFruitLists*(3, fruitList.getFruitList1(), 4,

fruitList.getFruitList2());

List<String> list2 = Arrays.*asList*(result3);

*yakshaAssert*(*currentTest*(), (list1.containsAll(list2) ? "true" : "false"), *businessTestFile*);

}

}

output\_revised.txt

testDeleteAndMergeFruitLists=true

testing-JavaMasterClassMqTwo.xml

<test-cases>

<cases xsi:type="java:com.assessment.data.TestCase">

<test-case-type>Functional</test-case-type>

<expected-ouput>true</expected-ouput>

<name>testDeleteAndMergeFruitLists</name>

<weight>10</weight>

<mandatory>true</mandatory>

<desc>Test to delete and merge fruit lists using string array and ArrayList</desc>

</cases>

</test-cases>

Test Data1

Enter size of first ArrayList

3

Enter fruit names

Apple

Cherry

Grapes

Enter size of second ArrayList

4

Enter fruit names

Orange

Mango

Melon

Apple

Merged fruitlist is

Cherry

Grapes

Orange

Learning outcome: Participant could able to learn string array and ArrayList.