**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit**

**Scenario:**

You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods

**Source Code:**

**StringUtil.java**

**package** com.example;

**public** **class** StringUtil {

**public** String reverse(String text) {

**return** **new** StringBuilder(text).reverse().toString();

}

**public** String capitalize(String text) {

**if** (text == **null** || text.isEmpty()) **return** text;

**return** text.substring(0, 1).toUpperCase() + text.substring(1).toLowerCase();

}

**public** **boolean** isPalindrome(String text) {

String cleaned = text.replaceAll("[^A-Za-z0-9]", "").toLowerCase();

**return** cleaned.equals(**new** StringBuilder(cleaned).reverse().toString());

}

**public** **long** countVowels(String text) {

**return** text.chars()

.filter(ch -> "aeiouAEIOU".indexOf(ch) >= 0)

.count();

}

}

**StringUtilAAATest.java**

**package** com.example;

**import** org.junit.After;

**import** org.junit.Before;

**import** org.junit.Test;

**import** **static** org.hamcrest.CoreMatchers.~~is~~;

**import** **static** org.junit.Assert.\*;

**public** **class** StringUtilAAATest {

**private** StringUtil util;

**private** **static** **int** *testCounter* = 0;

@Before

**public** **void** setUp() {

util = **new** StringUtil();

System.***out***.println("\n─── Test #" + (++*testCounter*) + " START ───");

}

@After

**public** **void** tearDown() {

util = **null**;

System.***out***.println("─── Test #" + *testCounter* + " END ───");

}

@Test

**public** **void** reverse\_returnsReversedString() {

String input = "JUnit";

String result = util.reverse(input);

assertThat(result, *is*("tinUJ"));

System.***out***.println("reverse(\"" + input + "\") = " + result);

}

@Test

**public** **void** capitalize\_capitalisesFirstLetter() {

String input = "hELLO";

String result = util.capitalize(input);

*assertEquals*("Hello", result);

System.***out***.println("capitalize(\"" + input + "\") = " + result);

}

@Test

**public** **void** isPalindrome\_detectsPalindrome() {

String word = "Madam";

**boolean** res = util.isPalindrome(word);

*assertTrue*(res);

System.***out***.println("\"" + word + "\" is palindrome? " + res);

}

**public** **void** countVowels\_countsCorrectly() {

String text = "JUnit in Action";

**long** count = util.countVowels(text);

*assertEquals*(6, count);

System.***out***.println("countVowels(\"" + text + "\") = " + count);

}

}

**Pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.example</groupId>

<artifactId>string-app</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>24</maven.compiler.source>

<maven.compiler.target>24</maven.compiler.target>

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

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

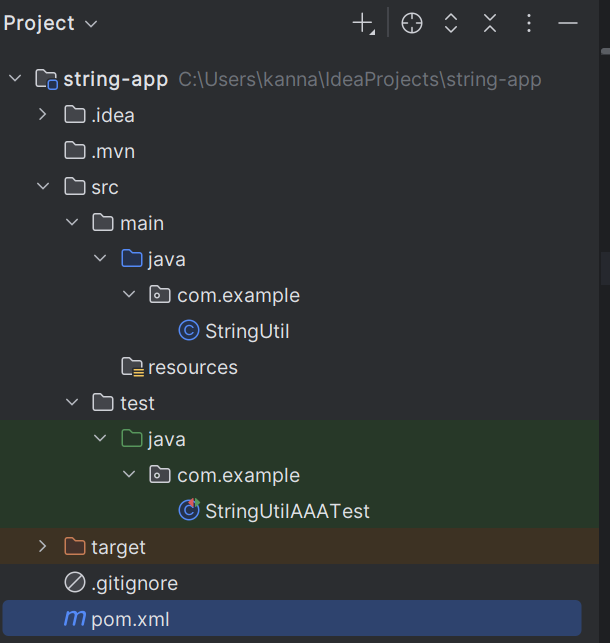
<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>



**Output:**

