Day 27

Assignment 3: Create test cases with assertEquals, assertTrue, and assertFalse to validate the correctness of a custom String utility class.

A)

Certainly! Here's a detailed explanation of the provided code:

StringUtils Class

This class provides utility methods for string manipulation: checking if a string is empty, reversing a string, and checking if a string is a palindrome.

Java code:

```
public class StringUtils {
    // Returns true if the string is null or empty
    public boolean isEmpty(String str) {
        return str == null || str.isEmpty();
    }
    // Reverses the given string
    public String reverse(String str) {
        if (str == null) {
            return null;
        }
        return new StringBuilder(str).reverse().toString();
    }

    // Checks if the string is a palindrome
    public boolean isPalindrome(String str) {
        if (str == null) {
            return false;
        }
        String reversedStr = reverse(str);
        return str.equals(reversedStr);
    }
}
```

- isEmpty(String str): Returns true if the input string is null or empty; otherwise, returns false.

- reverse(String str): Returns the reverse of the input string. If the input string is null, it returns null.

- isPalindrome(String str): Returns true if the input string is a palindrome (i.e., it reads the same backward as forward). If the input string is null, it returns false.

StringUtilsTest Class

This class uses JUnit 5 to test the methods in the StringUtils class.

Java code:

```
import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

@TestInstance(TestInstance.Lifecycle.PER_CLASS)
public class StringUtilsTest {

    private StringUtils stringUtils;

    @BeforeAll
    public void setUpBeforeClass() {
        System.out.println("Setup before all tests");
    }

    @AfterAll
    public void tearDownAfterClass() {
        System.out.println("Cleanup after all tests");
    }

    @BeforeEach
    public void setUp() {
        System.out.println("Setup before each test");
        stringUtils = new StringUtils();
    }

    @AfterEach
    public void tearDown() {
        System.out.println("Cleanup after each test");
    }

    @Test
    public void testIsEmpty() {
        assertTrue(stringUtils.isEmpty(null));
        assertTrue(stringUtils.isEmpty(""));
        assertFalse(stringUtils.isEmpty("abc"));
    }

    @Test
    public void testReverse() {
        assertEquals("cba", stringUtils.reverse("abc"));
        assertEquals("", stringUtils.reverse(""));
        assertEquals(null, stringUtils.reverse(null));
    }

    @Test
    public void testIsPalindrome() {
        assertTrue(stringUtils.isPalindrome("madam"));
        assertFalse(stringUtils.isPalindrome("hello"));
        assertFalse(stringUtils.isPalindrome(null));
        assertTrue(stringUtils.isPalindrome(""));  // Considering empty string as palindrome
    }
}
```

- *@BeforeAll public void setUpBeforeClass()*: Runs once before all tests to set up resources. Here, it prints a setup message.

- *@AfterAll public void tearDownAfterClass()*: Runs once after all tests to clean up resources. Here, it prints a cleanup message.

- *@BeforeEach public void setUp()*: Runs before each test to set up the test environment. Here, it initializes the stringUtils object and prints a setup message.

- *@AfterEach public void tearDown()*: Runs after each test to clean up the test environment. Here, it prints a cleanup message.

- *testIsEmpty()*: Tests the isEmpty method with different inputs:
  - null (should return true)
  - Empty string "" (should return true)
  - Non-empty string "abc" (should return false)

- *testReverse()*: Tests the reverse method with different inputs:
  - Non-empty string "abc" (should return "cba")
  - Empty string "" (should return "")
  - null (should return null)

- *testIsPalindrome()*: Tests the isPalindrome method with different inputs:
  - Palindrome string "madam" (should return true)

- Non-palindrome string "hello" (should return false)

- null (should return false)

- Empty string "" (should return true, considering an empty string as a palindrome)

Summary:

The StringUtilsTest class is designed to test the functionality of the StringUtils class methods. It ensures that each method behaves as expected with various input scenarios, including edge cases. The setup and teardown methods ensure that the test environment is properly initialized and cleaned up before and after each test, providing a clean state for each test execution.