

LAB20_ANPC6339_UNITTESTING_STRINGMANUPULATION

Due date: Thursday, 21 December 2023, 5:30 AM

Maximum number of files: 1

Type of work: Individual work

Lingabathula Thapaswi[AF0339471]

Assignment-1.

Create a JUnit test class to test a StringManipulator class that provides methods for manipulating strings.

Write parameterized tests to cover cases like reversing a string, converting to uppercase, and checking for palindrome strings.

Use parameterized tests to validate the string manipulation methods

Solution:-

Testcase→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>com.mypack</groupId>
  <artifactId>Testcase1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <dependency>
      <!-- https://mvnrepository.com/artifact/junit/junit -->
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>5.9.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-params</artifactId>
      <version>5.7.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
-----
```

Testcase1→src/main/java→com.StringManipulator→StringManipulator.java

```
package com.StringManipulator;
```

```
public class StringManipulator {
    public String toUppercase(String input) {
        return input.toUpperCase();
    }
    public boolean isPalindrome(String input) {
        StringBuilder reversed = new StringBuilder(input).reverse();
        return input.equalsIgnoreCase(reversed.toString());
    }
}
```

```

public String reverseString(String input) {
    return new StringBuilder(input).reverse().toString();
}
}

```

 Testcase1→src/test/java→com.StringManipulatorTest→StringManipulatorTest.java

```

package com.StringManipulatorTest;

import com.StringManipulator.*;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvFileSource;
import org.junit.jupiter.params.provider.CsvSource;
import org.junit.jupiter.params.provider.ValueSource;

public class StringManipulatorTest {

    @ParameterizedTest
    @ValueSource(strings = {"hello", "world", "JUnit"})
    void testToUpperCase(String input) {

        StringManipulator manipulator = new StringManipulator();

        String result = manipulator.toUpperCase(input);

        assertEquals(input.toUpperCase(), result);

    }

    @ParameterizedTest
    @ValueSource(strings = {"radar", "level", "deified"})
    void testIsPalindromeTrue(String input) {

        StringManipulator manipulator = new StringManipulator();

        boolean result = manipulator.isPalindrome(input);

        assertTrue(result);

    }

    @ParameterizedTest
    @ValueSource(strings = {"hello", "world", "java"})
    void testIsPalindromeFalse(String input) {

```

```
StringManipulator manipulator = new StringManipulator();

boolean result = manipulator.isPalindrome(input);

assertTrue(!result);

}

@ParameterizedTest

@CsvSource({"hello, olleh", "world, dlrow", "JUnit, tinUJ"})

void testReverseString(String input, String expected) {

StringManipulator manipulator = new StringManipulator();

String result = manipulator.reverseString(input);

assertEquals(expected, result);

}

}
```

Output:

Testcase1→src/test/java→com.StringManipulatorTest→StringManipulatorTest.java

Right click on StringManipulatorTest.java→new→run as→ 2 JUnit Test







