Lab 20 - Assignment

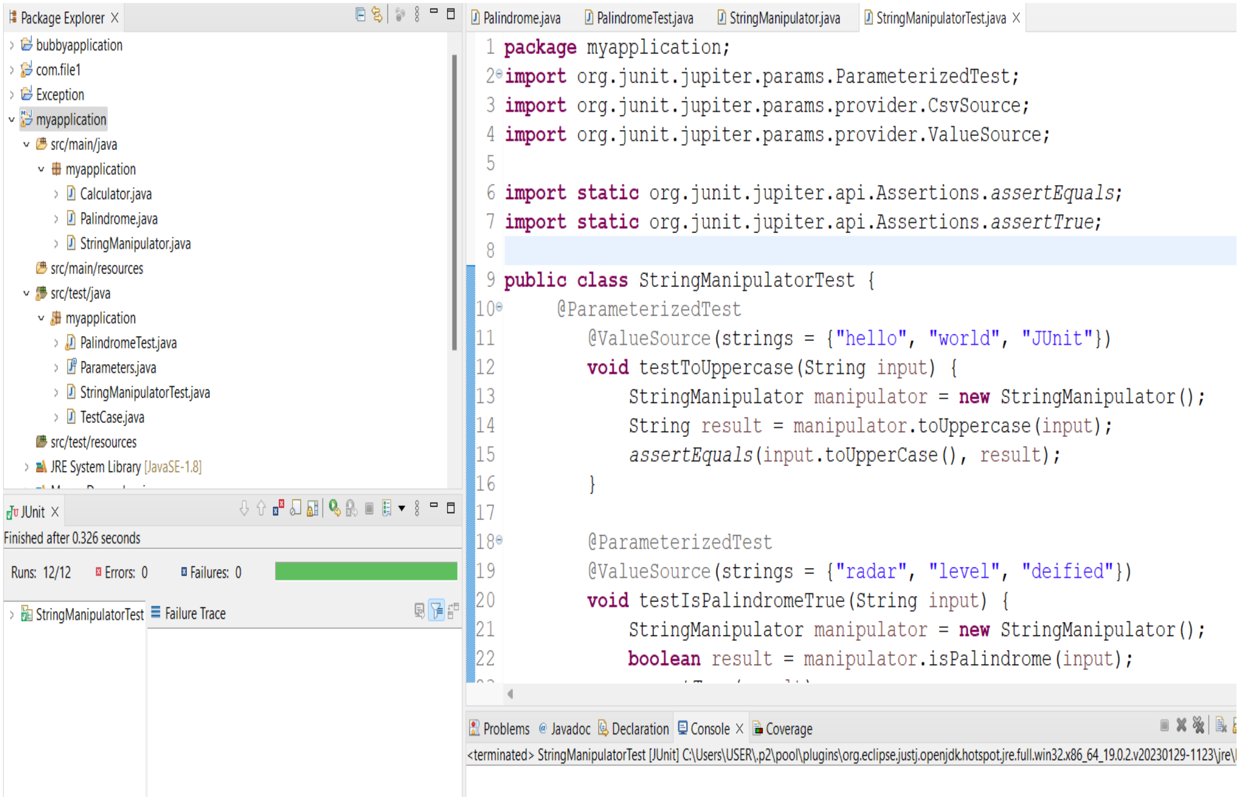
1) Create a JUnit test class to test a StringManipulator class that provides methods for manipulating strings. Write parameterized tests to cover cases like reversing a string, converting to uppercase, and checking for palindrome strings. Use parameterized tests to validate the string manipulation methods

```
package myapplication;

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();}


}


package myapplication;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;
import org.junit.jupiter.params.provider.ValueSource;

import static
org.junit.jupiter.api.Assertions.assertEquals;
import static
org.junit.jupiter.api.Assertions.assertTrue;
```

```
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:



Finished after 0.326 seconds

Runs: 12/12   Errors: 0   Failures: 0

```
package myapplication;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;
import org.junit.jupiter.params.provider.ValueSource;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTrue;

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);
```

```
<terminated> StringManipulatorTest [JUnit] C:\Users\USER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_19.0.2.v20230129-1123\jre\
```