

1. Reverse a text.

From my experience, reverse a text is a very popular coding task used during job interview. Text manipulation methods are the ones mostly used by programmers. Although you do not need to implement them, it is desired to understand how such operations are performed under the hood. The low-level details summed together have a significant impact on overall system performance.

Solution

The `String` class represents a text value in Java which is built from an array of characters. To implement a custom method to reverse a `String` you also have to operate on an array of characters. There are many solutions to reverse a `String`, but the most optimal should not allocate additional memory if there is no need. It is recommended to take an input `char[]` array, iterate through it and switch the first element with the last, the second element with the penultimate, etc. until half of the array is reached as presented in Listing 1.1.

Listing 1.1 – Reverse a text algorithm.

```
public class StringUtils {  
    public static String reverse(String input) {  
        if (input == null) {  
            return "";  
        }  
        char[] array = input.toCharArray();  
        final int halfLength = array.length/2;  
        int idx2;  
        char clipboard;  
        for (int idx1=0; idx1 < halfLength; idx1++) {  
            idx2 = array.length - 1 - idx1;  
            clipboard = array[idx1];  
            array[idx1] = array[idx2];  
            array[idx2] = clipboard;  
        }  
        return String.valueOf(array);  
    }  
}
```

Tests

The `StringUtils.reverse()` method is a good example of code which is easy to test with unit tests. Listing 1.2 presents an example implementation of such a unit test. In this case, a test is parametrized with two variables: `expected` and `input` values. The `data()` method, is a factory method which returns all combinations of input and expected data that need to be tested. Remember, preparing test cases and a method body simultaneously (TDD) results in high-quality code.

Listing 1.2 – Unit test used to verify the `StringUtils.reverse()` method.

```
@RunWith(Parameterized.class)
public class StringUtilsReverseTest {
    @Parameters(name = "{0}: reverse({1})={2}")
    public static Collection data() {
        return Arrays.asList(new Object[][]{
            {"", null},
            {"", ""},
            {"a", "a"},
            {"1234567890", "0987654321"},
            {"aBbA", "AbBa"}
        });
    }

    @Parameter(value = 0)
    public String expected;

    @Parameter(value = 1)
    public String input;

    @Test
    public void reverse() {
        assertEquals(expected, StringUtils.reverse(input));
    }
}
```