System Requirements Specification Index

For

String Manipulation and Information Operations

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

USE CASE DESCRIPTION

System Requirements Specification

1 Project Abstract

This project will evaluate the understanding of string manipulation and string information methods in Java. You will demonstrate their knowledge by declaring string variables and performing various string manipulation and information operations.

2 Assessment Tasks

Task 1:

1. **Declare 2 variables:**
   - A variable named `str1` of `String` datatype, initialized with the value `"Hello, World!"`.
   - A variable named `str2` of `String` datatype, initialized with the value `"Java Programming"`.

2. **Perform String Manipulation:**
   Use the string variables `str1` and `str2` to perform the following manipulations:

   - **Substring**:
     1) Extract a substring from `str1` starting at index `7` and ending at index `12`.
     2) Store the result in a variable named `subStr` of `String` datatype.
   - **Concatenation**:
     1) Concatenate `str1` with `" - "` and `str2`.
     2) Store the result in a variable named `concatenated` of `String` datatype.
   - **Replace**:
     1) Replace the word `"World"` in `str1` with `"Java"`.
     2) Store the result in a variable named `replaced` of `String` datatype.
   - **Convert to Uppercase**:
     1) Convert `str1` to uppercase using `toUpperCase()`.
     2) Store the result in a variable named `upperCase` of `String` datatype.
   - **Convert to Lowercase**:
     1) Convert `str2` to lowercase using `toLowerCase()`.
     2) Store the result in a variable named `lowerCase` of `String` datatype.

   Print the Results:
   - Print the results of each string manipulations i.e, `subStr`, `concatenated`, `replaced`, `upperCase`, and `lowerCase` with appropriate labels in separate lines as shown in the expected output.

Task 2:

3. **Declare a new string variable:**
    - A variable named `str` of `String` datatype, initialized with the value `"Java Programming"`.
4. **Perform String Information Operations:**

    Use the string variable `str` to perform the following information retrieval operations:

    - **Length**:
        1) Compute the length of the string using `length()`.
        2) Store the result in a variable named `length` of `int` datatype.
    - **Character at Index**:
        1) Retrieve the character at index `5` using `charAt(5)`.
        2) Store the result in a variable named `charAt` of `char` datatype.
    - **Index of Substring**:
        1) Find the starting index of `"Java"` in `str` using `indexOf("Java")`.
        2) Store the result in a variable named `indexOfJava` of `int` datatype.
    - **Check if String is Empty**:
        1) Check if the string is empty using `isEmpty()`.
        2) Store the result in a variable named `isEmpty` of `boolean` datatype.
    - **Check if String Starts With a Substring**:
        1) Check if the string starts with `"Java"` using `startsWith("Java")`.
        2) Store the result in a variable named `startsWithJava` of `boolean` datatype.
    - **Check if String Ends With a Substring**:
        1) Check if the string ends with `"Programming"` using `endsWith("Programming")`.
        2) Store the result in a variable named `endsWithProgramming` of `boolean` datatype.

    Print the Results:

    - Print the results of each operation i.e, `length`, `charAt`, `indexOfJava`, `isEmpty`, `startsWithJava` and `endsWithProgramming` with appropriate labels in separate lines as shown in the expected output.

Expected Output:

```
Substring: World
Concatenated: Hello, World! - Java Programming
Replaced: Hello, Java!
Uppercase: HELLO, WORLD!
Lowercase: java programming
Length: 16
Character at index 5: P
Index of 'Java': 0
Is empty: false
Starts with 'Java': true
Ends with 'Programming': true
```

3 Template Code Structure

3.1 Package: com.yaksha.assignment.StringManipulationInfoAssignment

Resources

Class/Interface	Description	Status
StringManipulationInfoAssignment (class)	• Main class demonstrating string manipulation operations such as: `substring`, `concat`, `replace`, `toUpperCase`, `toLowerCase`. • And string information operations like: `length`, `charAt`, `indexOf`, `isEmpty`, `startsWith`, and `endsWith`.	Need to be implemented.

4 Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) → Terminal →New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**

6. **To run your project use command:**
   mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.StringManipulationInfoAssignment"
7. **To test your project test cases, use the command**
   mvn test

8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**