System Requirements Specification Index

For

User Input using Scanner

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

USE CASE DESCRIPTION

System Requirements Specification

1 Project Abstract

User Input Using Scanner is a console-based Java application that demonstrates the usage of the Scanner class for taking user input. The program should take input of different data types (integer and string) from the user, perform an arithmetic operation on the integer, and display the results.

2 Assessment Tasks

1. **Declare and Initialize Scanner:**
   - Import the `Scanner` class from the `java.util` package to enable user input.
   - Create a `Scanner` object named `scanner` to take inputs from the user.
2. **Take Inputs from the User:**
   - **Integer Input**:
     1) Prompt the user with message: "Enter an integer:" using `System.out.print`.
     2) Take the input from user of type `int` datatype and store the input value in a variable named `userInputInt` using `scanner.nextInt()`.
   - **String Input**:
     1) Prompt the user with message: "Enter a string:" using `System.out.print`.
     2) Take the input from a user of type `String` datatype and store the input value in a variable named `userInputString` using `scanner.next()`.
3. **Perform Arithmetic Operation:**
   - Multiply the value of `userInputInt` by `2` and store the result in a variable named `result` of `int` datatype.
4. **Print the Results:**
   - Print the value of `result` with the label `"Double of your input integer:"`.
   - Print the value of `userInputString` with the label `"You entered:"`.
5. **Expected Output:**

   Enter an integer: 5

   Enter a string: Hello

Double of your input integer: 10

You entered: Hello

3 Template Code Structure

3.1 Package: com.yaksha.assignment.UserInputAssignment

Resources

Class/Interface	Description	Status
UserInputAssignment (class)	• Main class containing the logic to take user input using the Scanner class and process it.	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.UserInputAssignment"
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.**