
System Requirements Specification Index

For

Java Skills Evaluation

Version 1.0

USE CASE DESCRIPTION

Use Case 1

Write a program to take input of string from the user and count the number of uppercase alphabets, lowercase alphabets, digits and special characters. Store these counts in an array of size 4 in following order : UpperCase count, Lowercase count, Digit count, special char count

Example :

Input: "Test,String12!"

Output: [2, 8, 2, 2]

Description

1. Take console input as string, in **main ()** method, from user and store them in Integer List Collection
2. Pass the string to method **countDiffStringCharTypes()** and write the logic in that method to find total odd values.
3. We have considered array stores count of char in sequence of uppercase alphabets, lowercase alphabets, and digits.
4. Return the array having a number of uppercase alphabets, lowercase alphabets, digits and special characters from **countDiffStringCharTypes ()** to **main()** and display the count.
5. Template Code of the same is provided at:
com.iiht.evaluation.yaksha.string.CountStringCharacter

Use Case 2

Write a program to take input of date from the user in string format and display the day on which it falls. The code must fulfil following requirements:

- a. Method for validating that the integer-representing month is between 1 & 12
- b. Method for checking that the day part of the date objects is within the correct range for a month.
- c. Method for obtaining the week day from a given date.

Example:

Input: "08/09/2021"

Output: "WED"

Description

1. Take console input of a Date as "DD/MM/YYYY" in string format, in **main ()** method from user.
2. Pass the date to method **dateMonthValidation()** and write the logic in that method to find whether a given date contains a valid month from 1-12.
3. Return the Boolean from **dateMonthValidation()** to **main()**.
4. If **dateMonthValidation** returns true, call **isValidDay()**;
5. Pass the date to method **isValidDay()** and write the logic in that method to find whether the day in the given date is valid.
6. Return the Boolean result from **isValidDay()** to **main()**.
7. If **isValidDay()** is true, call **getWeekDay()**;
8. Pass the date to the method **getWeekDay ()** and write the logic in that method to find weekday on a given date.
9. Return the week day as String from **getWeekDay ()** to **main()**, as "MON" or "TUE" etc.
10. Template Code of the same is provided at:

com.iiht.evaluation.yaksha.datevalidation.DateOperation

EXECUTION STEPS

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. Command to run code for use-case 1:

```
mvn compile exec:java -  
Dexec.mainClass="com.iiht.evaluation.yaksha.string.CountStringCharacter"
```
7. Command to run code for use-case 2:

```
mvn compile exec:java -  
Dexec.mainClass="com.iiht.evaluation.yaksha.datevalidation.DateOperation"
```
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.

CONSIDERATIONS

1. **Mandatory:** Run the following command before the final submission

mvn test

(You can run this command multiple times to identify the test case status, and refactor code to make maximum test cases passed before final submission)

2. Your code will also be evaluated for code quality, naming conventions, readability etc.
3. **Make sure you do not modify existing class and method names and their signatures, else it would severely affect final result**
4. **Make sure you do not add any new class or methods, else it would severely affect final result**
5. **Make sure you do not modify any test cases, else it would severely affect final result**