COUNT SPECIFIC CHARACTER - PLAIN JAVA

Tuesday, April 4, 2017 9:48 PM

Tools/Technology Used:
JAVA (JDK 1.8)
ECLIPSE
GIT

TASK	TASK DESCRIPTION	EFFECTIVE LEARNING
1	Count the number of e (Case insensitive) in the below input Write a separate java method for performing the below: Input: "AccidEntally prEssed print screen" Output: 5 Just print the output inside the method itself No need of returning the output Execute the function from main method. Create the method as static	• Static methods • Basic string variable operations
2	Change the method as non static • Instead of initializing the input inside a method. • Create a class variable • Initialize that variable using constructor • Make use of that variable inside the method	• Class variables • Constructor • Non static method
3	Continuation from task2 Now initialize the class variable using setters and getters Add toString() method Now make use of the setter method to retrieve the variable for performing your operations	• Setters & getters • toString()
4	Instead of hardcoding the input, retrieve the user input and store it in setter Retrieve the variable from getters and make use of it	• Retrieving user input for word to search from
5	Now instead of just looking for the character 'e'. • Make sure that you modify your code to count for any character provided by the user as input	• Retrieving user input for character to count the number of occurrences for
6	Handle exceptions The function should execute in a loop till the user enters "y" OR "Y" If the user enters a "n" OR "N", it should stop the execution of the function	• Added looping to execute the task n number of times
7	Instead of creating same variable from the class,	• Create a separate POJO class

	create a POJO class * Store the variable in setters of that POJO class * * * Make use of that POJO in task7 class * * Main method is heavily loaded * * Whatever logic that has been written inside main, move * into a separate new method * * * From main, just call this method * * Added JAVA Docs	• Javadocs • Segregating different logic under different methods
8_1	* Rewrite the string logic using hashmap * * created separate methods for fetching inputs, business logic. Implemented the core logic to be executed multiple times	• Usage of hashmap
8_2	Add the value of userCharcToSearch in POJO using setters & getters *Return the output in the innerList of lists * *Example: *Input String: cleiii , Char to search:i *Input String: viveevv , Char to search:v * *Output should be like this: (('cleiii','i','3') , ('viveevv','v','4')) * * *Return list of list as the output for the method	• List of list
8_3	*Return the output in the below format: * *Example: *Input String: cleiii , Char to search:i *Input String: viveevv , Char to search:v * *Output should be like this: (<1, ('cleiii','i','3')> , <2,('viveevv','v','4')>) * *<> indicates map *() indicates list * * *Return list of list as the output for the method	• Collections - Map & List combined
8_4	Except countNumberOfCharacters() keep all business *logic should be present in separate class	• Separating out the business logic

8	Rewrite the string logic using hashmap	• Basic HashMap
8_5	Refactoring the code. Introduce a controller class. Basic exceptional handling	• Refactoring the code
9	Reading input from a file and writing output to a file	• File Handling

Proposed Future Implementations (Project URI: https://github.com/vivek-muralidharan/CountSpecificCharacters-SpringBootProject **)**

- [ ] Convert it into a spring boot maven project
- [ ] Integration with MongoDb
- [ ] Perform input validations, Exceptional handling
- [ ] Add additional scenarios in REST
- [ ] Add additional business cases itself
    - Support for Reading/Writing XML
    - Support for Reading/Writing EXCEL
- [ ] JAVA 8 CONCEPTS (Streams, Lambda)
- [ ] Using polymorphism , abstract, overriding concepts
- [ ] Unit testing (Junit + Mockito +Coverage)
- [ ] Possible Integration testing
- [ ] Integration to Angular Frontend
- [ ] Building Project using frontend
- [ ] Data structure concepts - Binary search, constant time
- [ ] Possible Threading scenario