SWE 437-001

Software Testing and Maintenance

Assignment 3: Add Quotations Feature

CLI (Command Line Interface)

*Students:*

Behrad Behmardi

Pisith Yim

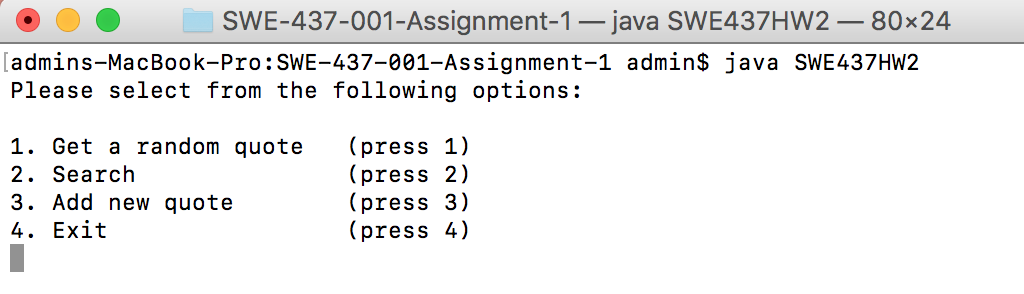
*Professor:*

Dr. Jeff Offutt

Git Repository: http://github.com/behmardibehrad/SWE-437-001-Assignment-1.git

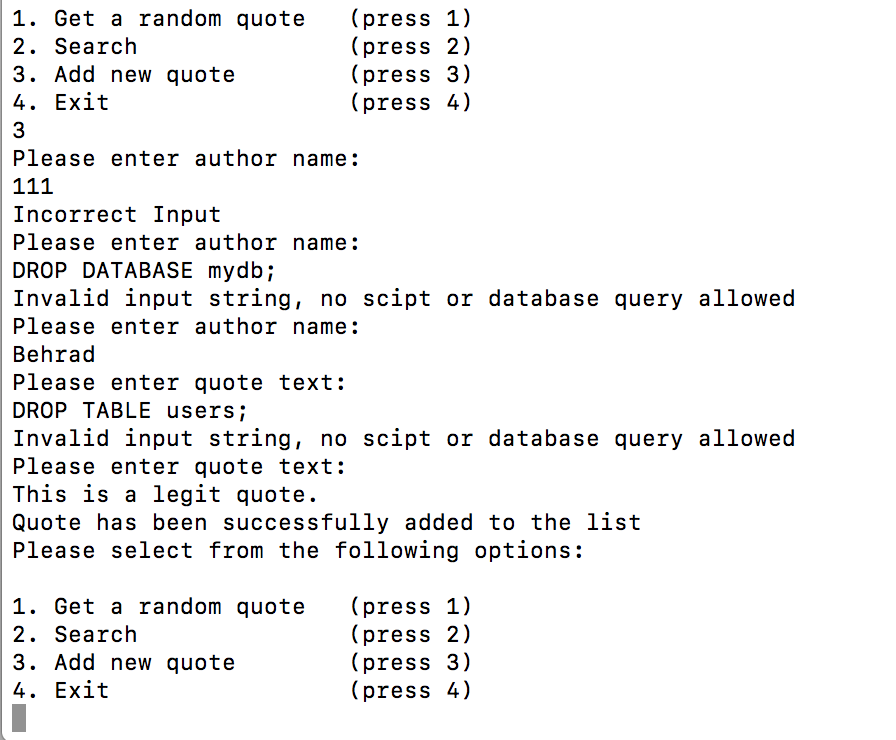
**Screen Shots verifying the functionality:**

**Main menu**

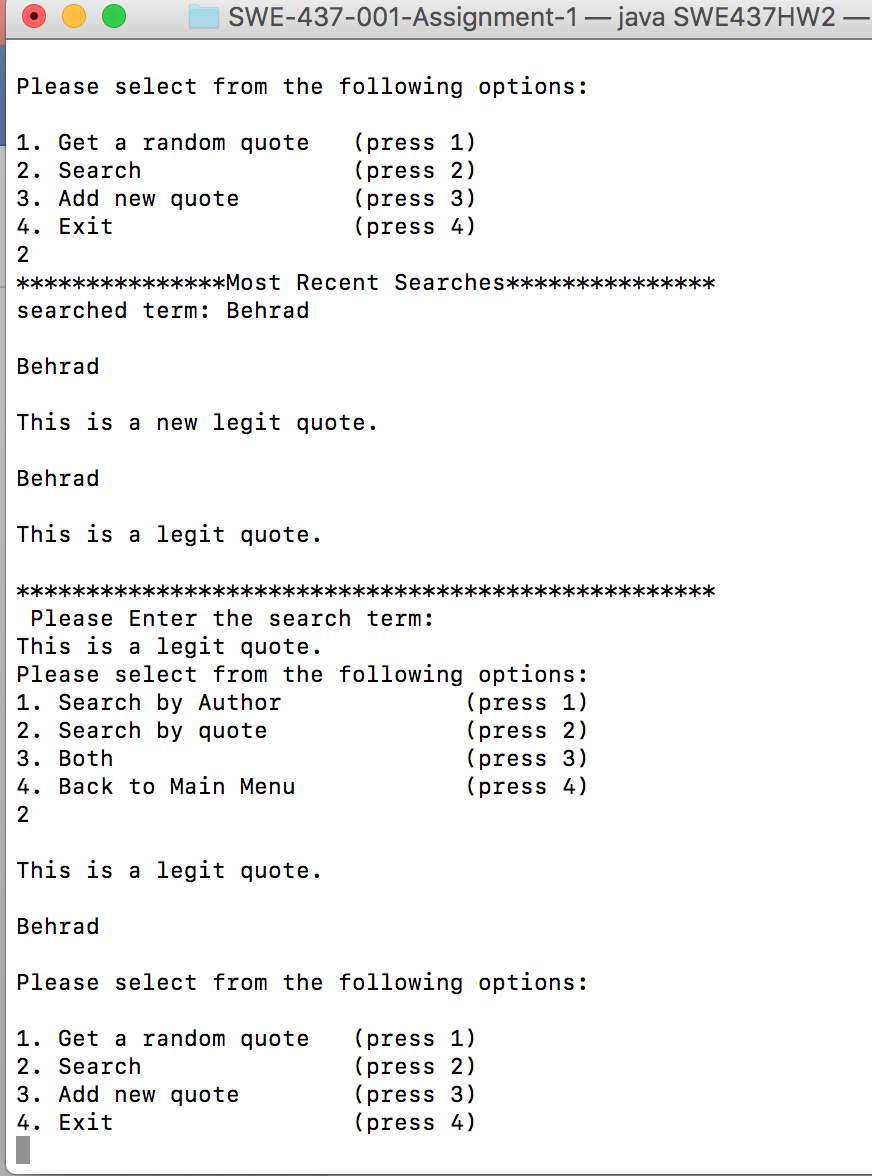


**Add quote (validation and integrity check)**

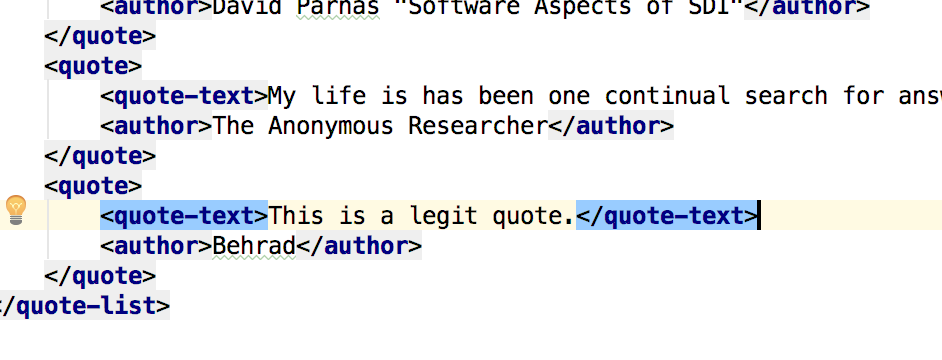
**Rules: no numbers only, empty string, or database query**

****

**Retrieve the newly added quote from quotes.xml**

****

**quotes.xml**

****

**Documentation Log:**

|  |  |  |
| --- | --- | --- |
| Date | Item | Change |
| 02/10/2018 | void addQuoteMenu() | New Function Created |
| 02/10/2018 | boolean addQuote(String author, String quoteText){} | New Function Created |
| 02/11/2018 | boolean checkErrorInput(String text){} | New Function Created |
| 02/11/2018 | boolean saveToXml(String xmlFile, QuoteList pList){} | New Function Created |

Implemented Functions in SWE437HW2.java:

* public static void addQuoteMenu(){}
  + Asks for the author’s name and the quote text to be entered by the user. First asks for author’s name and after the user enters the input, calls checkErrorInput(input) to validate the integrity of the input, if the input is not valid then the function enters a while loop and continuously asks for the correct user input. This sequence is true for quotes text as well. If both author’s name and quote’s text are validates then the functions calls addQuote(author, quoteText) function and passes the validated author’s name and quote’s text as inputs.
* public static boolean addQuote(String author, String quoteText){}
  + Creates a new quote using the user’s input and adds the newly created quote to the quote list. Finally calls the saveToXml(String xmlFile, QuoteList pList) and writes the quote to the targeted xml file. Returns the return value of saveToXml function meaning if write was successful returns true else returns false.
* public static boolean checkErrorInput(String text){}
  + Checks for the passed in string to be valid, it cross checks the string against number of known vulnerable strings and character’s pattern. So far this function checks that the string is not empty, is not a sql query, is not a shellcode, is not digits and more test case cold be easily added in future. Return true if string is invalid and returns false if string is valid.
* public static boolean saveToXml(String xmlFile, QuoteList pList){}
  + The purpose of this function is to convert the new updated list (with new code added), and append it to the xml file. We first we need to find the root of the xml hierarchy, which is quote-list in this case. We then create the quote, author, and quote-text elements. We then set the author and quote-text content. We can then append author, quote-text to quote, then append quote to quote-list. Finally, we write it back to xml. It will return true if everything executes successfully.

GitHub Commit History Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Commit ID | User | Date | Comment |
| 60d0270 | Seth | 02/02/2018 | added pics |
| 266f134 | Behrad Behmardi | 02/02/2018 | editing Readme.md |
| 6b37968 | Seth | 02/02/2018 | editing Readme.md |
| ff76876 | Behrad Behmardi | 02/02/2018 | editing Readme.md |
| 256d49b | Seth | 02/02/2018 | editing Readme.md |
| bce3458 | Behrad Behmardi | 02/02/2018 | Merge branch 'master' of https://SWE-437-001-Assignment-1 |
| 0acb4de | Seth | 02/02/2018 | editing Readme.md |
| 834fd76 | Seth | 02/02/2018 | Add new changes |
| 8cd472d | Seth | 02/02/2018 | Merge changes |
| db1dfee | Seth | 02/02/2018 | New maintainability text |
| 36ad572 | Behrad Behmardi | 02/02/2018 | Update maintainability.md |
| 78c22b1 | Seth | 02/02/2018 | New maintainablility text |
| 4b04ccb | Seth | 02/02/2018 | Add blockquote text |
| ee152d3 | Seth | 02/02/2018 | Changed maintainability.txt to .md |
| 4c2b2de | Seth | 02/02/2018 | Add maintainablilty.txt |
| f3f50b0 | Behrad Behmardi | 02/02/2018 | Delete quoteserve.class |
| bcef17c | Behrad Behmardi | 02/02/2018 | Delete SWE437HW2.class |
| 706c376 | Behrad Behmardi | 02/02/2018 | Delete QuoteSaxParser.class |
| 7469622 | Behrad Behmardi | 02/02/2018 | Delete QuoteSaxHandler.class |
| eab9019 | Behrad Behmardi | 02/02/2018 | Delete Quote.class |
| e17c50b | Behrad Behmardi | 02/02/2018 | Delete QuoteList.class |
| d9d7832 | Behrad Behmardi | 02/02/2018 | organized the documentation |
| adc9e50 | Behrad Behmardi | 02/02/2018 | deleted package name |
| 74d5b92 | Behrad Behmardi | 02/02/2018 | cleaned up and documentation |
| 1b49533 | Behrad Behmardi | 02/02/2018 | fixed comments for PrintSearchMenu() |
| a99205a | Seth | 02/02/2018 | Add switch statement |
| c6f25d3 | Behrad Behmardi | 02/01/2018 | added the scanner and menu options for PrintSearchMenu() |
| df26713 | Seth | 02/01/2018 | Add most recent searches layout |
| 0d16040 | Behrad Behmardi | 02/01/2018 | created function PrintSearchMenu() |
| 31f3d01 | Seth | 02/01/2018 | Create generateRandomQuote method |
| 7839528 | Behrad Behmardi | 02/01/2018 | finished userAnswer(int) function |
| 9d6bebd | Seth | 02/01/2018 | Completed printSearch method |
| b083393 | Behrad Behmardi | 02/01/2018 | created userAnswer(int) function |
| c8903eb | Seth | 02/01/2018 | create printSearch method |
| 7001c69 | Behrad Behmardi | 02/01/2018 | Added comments |
| 3ec44e5 | Seth | 02/01/2018 | Delete e from printMenue method |
| 3d89fb4 | Behrad Behmardi | 02/01/2018 | Added comments |
| 221dd89 | Behrad Behmardi | 02/01/2018 | Create printMenu method |
| 1f30b6e | Behrad Behmardi | 02/01/2018 | Merge branch 'master' of https://github.com/behmardibehrad/SWE-437-001 |
| 9ad9b2d | Behrad Behmardi | 02/01/2018 | Created 'SWE437HW2.java' |

**Description of the data integrity checks:**

We have implemented a function called **public static boolean checkErrorInput(String text)**

This function is responsible for data integrity check through out the code. Assuming that the user wants to add a new quote to the database or xml file, first the user is greeted with a menu that allows him/her to do so. User is asked to enter the author’s name, and the program will save the user’s input and pass it to the **checkErrorInput** function. Inside the function we are checking that the data is acceptable before adding it to the xml file, the list explains these checks:

1. Text is not a string of digits such as 123456789 etc
2. Text is not empty
3. Text is not more than 500 characters
4. Text is not sql queries
5. Text is not a script
6. Text is not a shellcode

If text is an actual legitimate quote then the function will return false, meaning that there was no error found within the text otherwise the function will return true indicating that there was a problem with the text and it will print out the message related to the error, for example if the text was an sql query then the program will print “no script or database query allowed”.

**Maintainability:**

As we have mentioned in previous maintainability assessment we will focus on creating small components that are easy to reuse throughout the code. In this assignment with having that in mind, we have created four new functions that all of them have the capability to be reused for different sections of the code in future. For example, the checkErrorInput(String) could be utilized in future for data integrity check of any user input, and all is needed by the programmer is to pass the targeted string as an input to this function and the function will return a Boolean (false = no error, and true = error). It is notable that adding these four new functions was fairly a simple task since during the last modification we have created the menus and other functionalities such that future requirements could be added easily. Another example would be the saveToXml(String xmlFile, QuoteList pList) which will take an xml file and quote list as input and add the quote to the xml file. Understanding that the second element passed to this function is a specific object may result is some confusion for reusability of tis function, however the function’s body could be easily modified to behave as expected for future requirements and improvements .