Project Stage 5: Bible Model and Basic GUI

You will *work in pairs that are assigned by me* on this stage. You should practice ***pair programming*** while completing this project. This means that you will work together at a single computer and take turns at the keyboard. Make sure that you don't dominate the effort, but also that you don't let the other person do everything. You need to both take equal responsibility for understanding the code you are writing. It is your responsibility to coordinate with each other and get the assignment completed on time.

In this stage you will implement the first version of the Bible Reader application. It will allow the user to search for verses with a given word and display in a sensible format (e.g. one per line) the verses that contain that word from a single version of the Bible. For inspiration on what your GUI might look like, you should run my sample solution (https://cusack.hope.edu/Notes/?Instructor=BibleReader). It is important to realize that this sample is from a further stage of the project so yours will not have all of the features you see in that example. It also performs searches differently so do not compare your results with this one.

1. Add fields to **BibleReaderModel** and implement the methods listed for Stage 5.
2. Notice that **BibleReaderApp** already calls the method to read the file in and creates and connects several of the key classes. It also contains the main method.
3. The main thing you need to do in **BibleReaderApp** is implement the GUI. This will consist of input (whatever components you want to use) and output (an instance of **ResultView**).
4. You are free to choose exactly what your GUI looks like, but it should be intuitive. For the input, the obvious choices are to use a textfield and button (probably the most sensible) or a JMenuItem with a dialog box.
5. The **ResultView** class displays the search results, and should include summary stats (e.g. "There are 39 verses containing the word *cellar*"). The summary stats should be in a separate component than the results.
6. The best choice for **ResultView** is probably a JEditorPane on a JScrollPane for the results (format the results using simple HTML tags) and a JTextField or JTextArea for the stats.
7. Run the Stage 5 tests and fix any errors until you pass all of the tests.
8. As usual, update the documentation (make sure to include both names in the files) and clean up your files before submitting them.
9. Once you are happy with your first version of Bible Reader, zip up the ***src*** folder that is in your project directory. *Please do not zip up your entire project directory since I do not need all of your copies of the fairly large Bible files! Only zip up the* ***src*** *directory!*
10. As usual, create **MyGrade_P5.txt** (copy it from a previous one) and fill in your actual time spent and expected grade (keep these on the same line as the headings!) and include a brief justification of your expected grade.
11. One person from each pair should hand in the zipfile and **MyGrade_P5.txt** using Handin under assignment **235-P5**. You should receive an e-mail shortly after submitting it with the test results.
12. Grades will be based on: Your GUI works as specified, the layout and formatting of the output is reasonable (buttons/textfields don't stretch/fill large areas, there are borders

and/or spaces between things, etc.), your implementation choices are reasonable, and your code is neat, organized, and well-documented.

13. Make sure your partner gets copies of all of the files when you have finished the project.

Hints/Comments:

- You should not change the signature of any of the methods that are already in the code.
- You should probably not be making anything (variables, fields, methods, etc.) **static** or **final**. If you are doing so, you are likely not having the classes properly interact with each other.
- Although you only need 1 version of the Bible this time, it might be wise to plan ahead and implement the model so it can store multiple versions of the Bibles and results.
- You can use a **JEditorPane** to display your results using simple HTML to format it nicely. You need to set the content type to "text/html". Also, you should set the caret position to 0 after you call ***setText***. This will make sure that the scrollpane scrolls back to the top (I am assuming you put the JEditorPane on a JScrollPane).
- Look at the Java APIs and/or Swing Tutorials to learn more about ***JScrollPane***, ***JEditorPane***, etc.
- For a sample HTML table that you can mimic in your JEditorPane, go to the ***Misc*** section of the ***Notes*** page (linked from the course website) and take a look at ***sampleHTMLTable.html***.