Alex Braverman

Guided Project

Questions:

1) The Dictionary allows you to save the current state to a file. What is the name of the file

that this program saves to your hard drive?

dictionary.json

2) What is the file extension of the file that is saved?

.json

3) Where is it located on your hard drive? ​ Hint:​​ There is a special name for the folder we

are saving to.

The file is saved in the user home directory

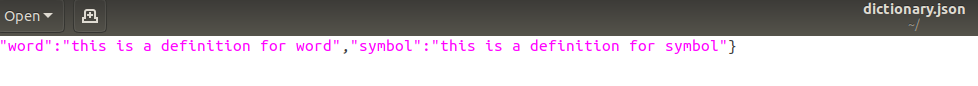
/home/alex

4) Run the program and save a few word/definitions pairs to file. Open the file with a

development-friendly editor (e.g., Notepad++). Technically, you can use IntelliJ by going

to File > Open. Take a screenshot of the opened file and paste it along with your

solutions to the questions above.



Bug 1

the Dictionary.findDefnition() method would return the value of definition as null when a word that wasn’t in the dictionary would be entered. To avoid this problem I added an if statement that would assign definition = “” if it is null after the first if statement. Now it even works with none string types like an int.

Bug 2

This bug was stuck in an infinite loop because validEntry was never set back to true. I just added an else statement to if(!dictionary.addEntry(word,definition)) setting it back to true and breaking the loop.

Bug 3

Used the logic in the boolean statements from addEntry and created a private boolean method that takes a string parameter then returns true if it’s a word or false if it’s not.

Design Issue 1

added throws IOException to convertMapToJsonString() then had to use a try/catch block saveFile() method in Dictionary.java.

Design Issue 2

to solve this I added Map<String, String> mapToStorageService as parameter to the JSONStorageService .saveFile() method and in all other effected places. I got rid of the try/catch inside the Dictionary.saveFile() method because it was no longer catching an exception since I’m calling the JSONStorageService .saveFile() which will try/catch the JSONStorageService.convertMapToJsonString() method. Now both the JSONStorageService .saveFile() and JSONStorageService .loadFile() method will both handle the conversion behind the scenes. For the test I had to add a new mockMapToStorageService HashMap to the DictionaryTest.saveFileShouldReturnTrue() method, it’s just an empty hashmap.

X3042

Feature 1

At first I thought this would be a simple copy/paste the code and I’m done, but of course it wasn’t. First I would have to use extends keyword for the new XMLDictionaryDemo there was no point in rewriting or coyping all that code, but I had to make a private FILE\_NAME variable which would save as “dictionary.xml” instead of json. This also required modification of the private methods in the parent class, I changed them to protected so that the children could use them as well.

I had to google and figure out which dependency I needed to add to the pom file in order to convert to xml like we did with json. After that everything seemed ok until I ran the xml demo and found that it wasn’t saving or loading because I was still using the JSONStorageService (inheritance mistake on my part.) Then I re-wrote the code for the XMLStorageServices. Instead of copy/paste I used extends on the JSONStorage Services to inherit the methods from that parent class then I also used implements for the StorageServices since I would need to make a new saveFile/loadFile method for .xml extensions because using the inherited methods wouldn’t work. This also required making new private methods for convertXMLStringToMap and convertMapToXMLString. To make sure that both xml and json would work I created a new parameter in the Dictionary constructor to accept a String fileType that would be passed in replacing this

String fullPathToFile = System.getProperty("user.home") + File.separator + fileType;

where fileType replaces “dictionary.json” so now it would be able to support any file type as long as it is passed in as a string argument to the constructor.

Also I removed the code in the do/while loop and put that into a method just to make it easier to reuse in my XML class, the method is used in a do/while loop still.