**Project 5 Write Up: Dolev Peleg**

**Approach, Design, and Algorithm**

This assignment was easier for me than our past projects. Because when I prepared for exam 2 I put a lot of time researching and working with trees, this project did not challenge me as much as the others did.

I started this project by first designing the most basic class, TreeNode, then I created MorseCodeTree, and lastly, MorseCodeConverter. Designing TreeNode and most of the methods of MorseCodeTree was simple. The only “tricky” methods were addNode, insert, fetchNode, fetch, and LNRoutputTraversal.

First, when I tried to add nodes using the AddNode method, I encountered some exceptions when trying to add a lower tree node when one of its parents was still not initialized. For example, I could not add ‘a’, because the node that is supposed to contain ‘e’, the parent node, was still null. I fixed this issue by creating new child nodes to null parents when needed, and setting the null parents to their correct value later on. This happened in lines 107 and 118 in my MorseCodeTree class.

Another issue that I encountered was that I could not operate on the tree without creating an instance of MorseCodeConverter. My first approach was to create a default constructor for MorseCodeConverter, and within it to call the MorseCodeTree constructor and buildTree method. This was incorrect because when I read the instructions again, I noticed that the methods of MorseCodeConverter are static, and it has one static MorseCodeTree field. I fixed it by adding a call to the buildTree method within the MorseCodeTree default constructor. Thus, everytime a MorseCodeTree is created, it automatically builds a morse code tree. This enabled me to be able to use the MorseCodeConverter in a static manner.

Another small issue that I had to fix was that when reading codes from text files, my scanner did not get any of the spaces in the file, thus causing the code to represent a really long string. After some testing, I realized this happened because I tried to get each character from the text file individually. I fixed it by using nextLine() method instead of next().

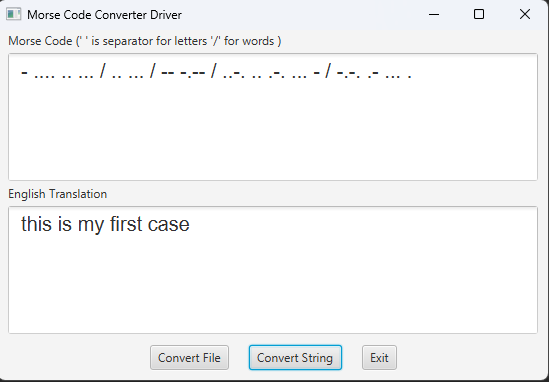
I also got confused with how to create an inorder traversal method. This is because although we worked a lot with trees, we never had to code traversal methods before. I found a website with the 3 lines of code that I was missing to create my working inorder traversal method. The lines of code that I used that website for are lines 264-266 in my MorseCode tree class. The website that I used will be cited below.

The rest of the project went smoothly, and I did not encounter any more issues.

**Test Runs and Cases**

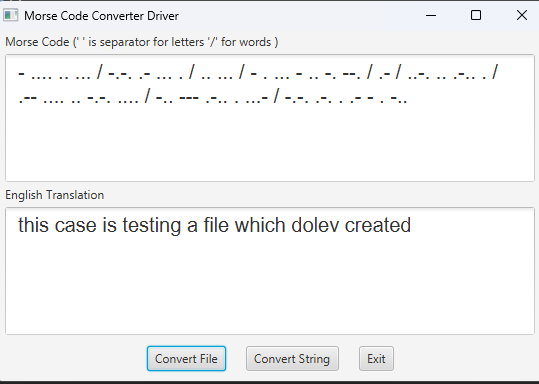
**Case 1:**

| Input | Expected Output | Actual Output |
| --- | --- | --- |
| - .... .. ... / .. ... / -- -.-- / ..-. .. .-. ... - / -.-. .- ... . | this is my first case | this is my first case |

****

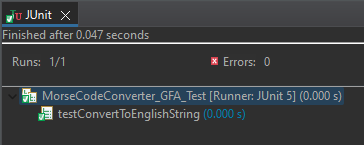
**Case 2(from a file):**

| Input | Expected Output | Actual Output |
| --- | --- | --- |
| - .... .. ... / -.-. .- ... . / .. ... / - . ... - .. -. --. / .- / ..-. .. .-.. . / .-- .... .. -.-. .... / -.. --- .-.. . ...- / -.-. .-. . .- - . -.. | this case is testing a file which dolev created | this case is testing a file which dolev created |

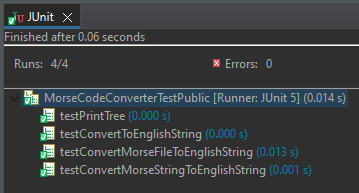
****

**JUnit Testing**

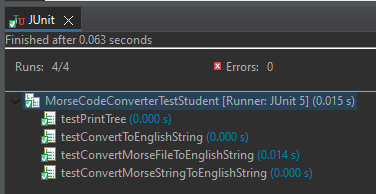
**MorseCodeConverter\_GFA\_Test**

****

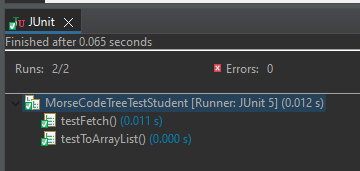
**MorseCodeConverterTestPublic**

****

**MorseCodeConverterTestStudent**

****

**MorseCodeTreeTestStudent**

****

**Learning Experiences**

Most of the work we did regarding trees in the class so far did not involve coding, therefore this was my first time coding trees.

I learned how to build a tree, and how to traverse it using inorder traversal. I also learned how to fetch and add nodes to a tree, and how to test tree classes.

Although I did not struggle with this project, it taught me a lot about trees, and I now feel more comfortable coding them.

All my output was as expected, and besides my notes and our textbook, I only used one resource in order to figure out how to traverse my tree using inorder traversal. This website is cited below.

**Assumptions**

1. The user will be using JUnit 5, Java, and JavaFX
2. Input in text files will follow the same format that was in the text file that was provided to me (one line per file).
3. On the provided write up, it said that MorseCodeTree is a generic class, but the JavaDoc builds it with a string argument rather than a generic argument. I followed the JavaDoc, therefore MorseCodeTree is not a generic class (this did not impact the project’s output).

**Enhancements**

No enhancements were made.

**Citations**

How can I return the result of a binary tree in traversal order in arraylist correctly? Quora. (n.d.). Retrieved April 22, 2023, from https://www.quora.com/How-can-I-return-the-result-of-a-binary-tree-in-traversal-order-in-ArrayList-correctly