Project 5: Morse Code Converter

Mohammad A. Kazemivarnamkhasti

Montgomery College

CMSC204, CRN 22443

UML Diagrams

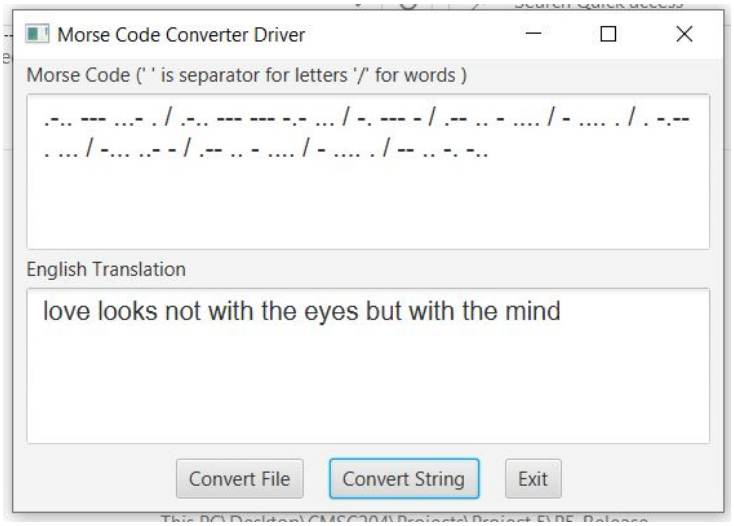
TreeNode<T>
–data: T –leftChild: TreeNode<T> –rightChild: TreeNode<T>
+TreeNode(T dataNode) +TreeNode(node: TreeNode<T>) +getData(): T

MorseCodeTree Implements LinkedConverterTreeInterface<String>
~treeNode: TreeNode<String> ~letters: ArrayList<String>
+MorseCodeTree() +addNode(root: TreeNode<String>, code: String, letter: String): void +buildTree(): void +toArrayList(): ArrayList<String> +LNRoutputTraversal(root: TreeNode<String>, list: ArrayList<String>): void +delete(data: String) throws UnsupportedOperationException: MorseCodeTree +update() throws UnsupportedOperationException: MorseCodeTree +fetch(code: String): String +fetchNode(root: TreeNode<String>, code: String): String +getRoot(): TreeNode<String> +setRoot(newNode: TreeNode<String>): void +insert(code: String, result: String): MorseCodeTree

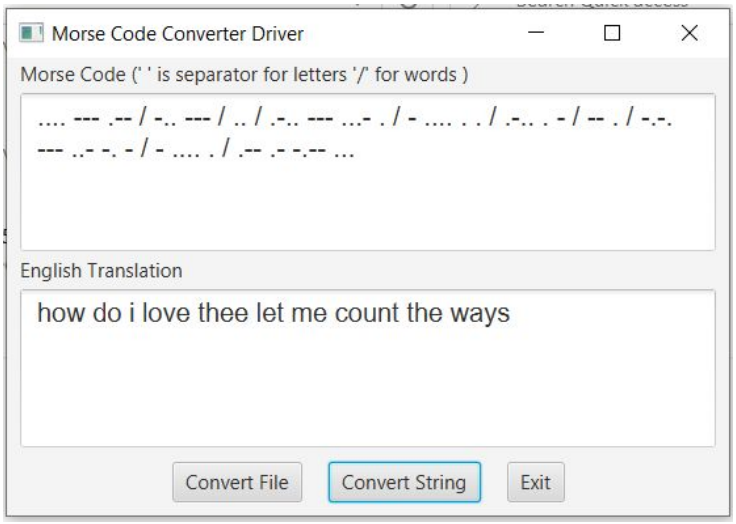
MorseCodeConverter
~mct: MorseCodeTree
+MorseCodeConverter() +convertToEnglish(codeFile: File) throws FileNotFoundException: String +convertToEnglish(String: code): String +printTree(): String

Screenshots of Test Cases

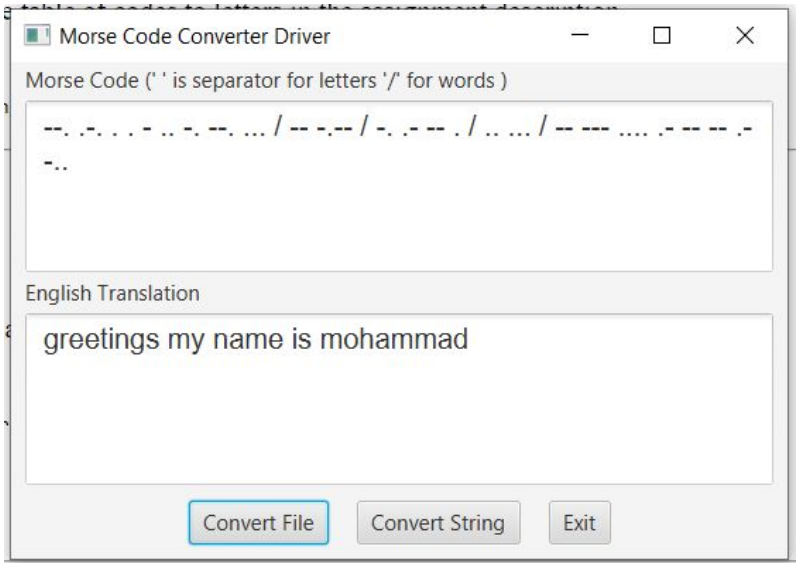
Case 1 using “Convert String” (Morse code from LoveLooksNot.txt)



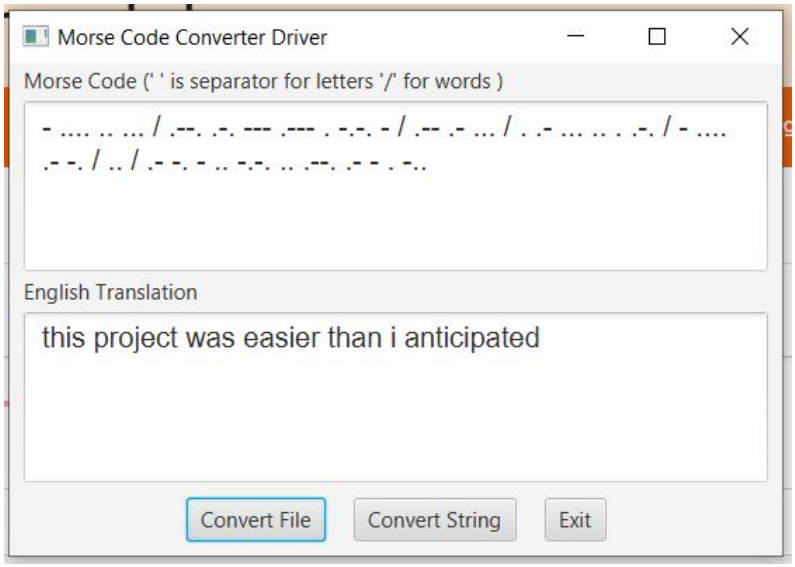
Case 2 using “Convert String” (Morse code from howDoILoveThee.txt)



Case 1 using “Convert File” (greetings.txt)



Case 2 using “Convert File” (anticipation.txt)



Learning Experience/Lessons Learned

The start of this project was quite bumpy. I was confused and even though I had some theories for how to approach it, I just couldn't wrap my head around it. Initially, I was under the impression that we were to make a set of MorseCodeTree's, similar to how we made a LinkedList of BasicDoubleLinkedLists's (in Project 3). After giving it more thought and looking through the possibilities, I realized that MorseCodeTree itself is the collection [of TreeNodes]. Once I confirmed this, the project became doable.

I learned quite a bit from this project. Although Project 3 made me more comfortable with using nodes and linked nodes, I wasn't nearly as comfortable with them as I am now. Another key area where I feel my competence has improved is recursion and recursive methods. At first, I wasn't sure how to store the String that was being returned from the recursive call in the *fetchNode* method of MorseCodeTree. It took some debugging and trial-and-error, but the solution was a simple assignment statement (line 118 of MorseCodeTree).

Other than that, I felt that there was some unnecessary repetition in this project. I feel that having separate methods for the operations of *fetch* and *fetchNode*, as well as *insert* and *addNode*, is repetitive and simply extra.