**Montgomery College Name: Kabindra Raj Suwal**

**CMSC 204**

**Assignment 6**

**Pseudocode**

**TreeNode:**

* Create a public TreeNode(T dataNode) constructor
* set left and right child to null and data to the dataNode
* create another constructor TreeNode(TreeNode<T> node) to make deep copies of the node
* create a public getData() method to return the data in TreeNode

**MorseCodeTree:**

* implements LinkedConverterTreeInterface
* Create a root
* Create a constructor MorseCodeTree() which calls buildTree() method.
* Create getRoot() method which returns reference of the root
* Create setRoot() method to set root of MorseCodeTree
* Create MorseCodeTree insert method to add elements in the tree
  + This method calls addNode()method
* Create addNode() method that adds to the tree
* Create fetch() method to fetch elements from the tree
  + This method calls fetchNode()method
* Create fetchNode() method that fetches elements from the tree
  + ‘.’ (dot) means to traverse left
  + ‘-‘ (dash) means to traverse right
* Create buildTree() method that builds MorseCodeTree by inserting nodes in specified places
* Create toArray() method that returns array list of items in linked tree
* Create LNRoutputTraversal() method that puts the contents of the tree in an ArrayList in LNR (Inorder)

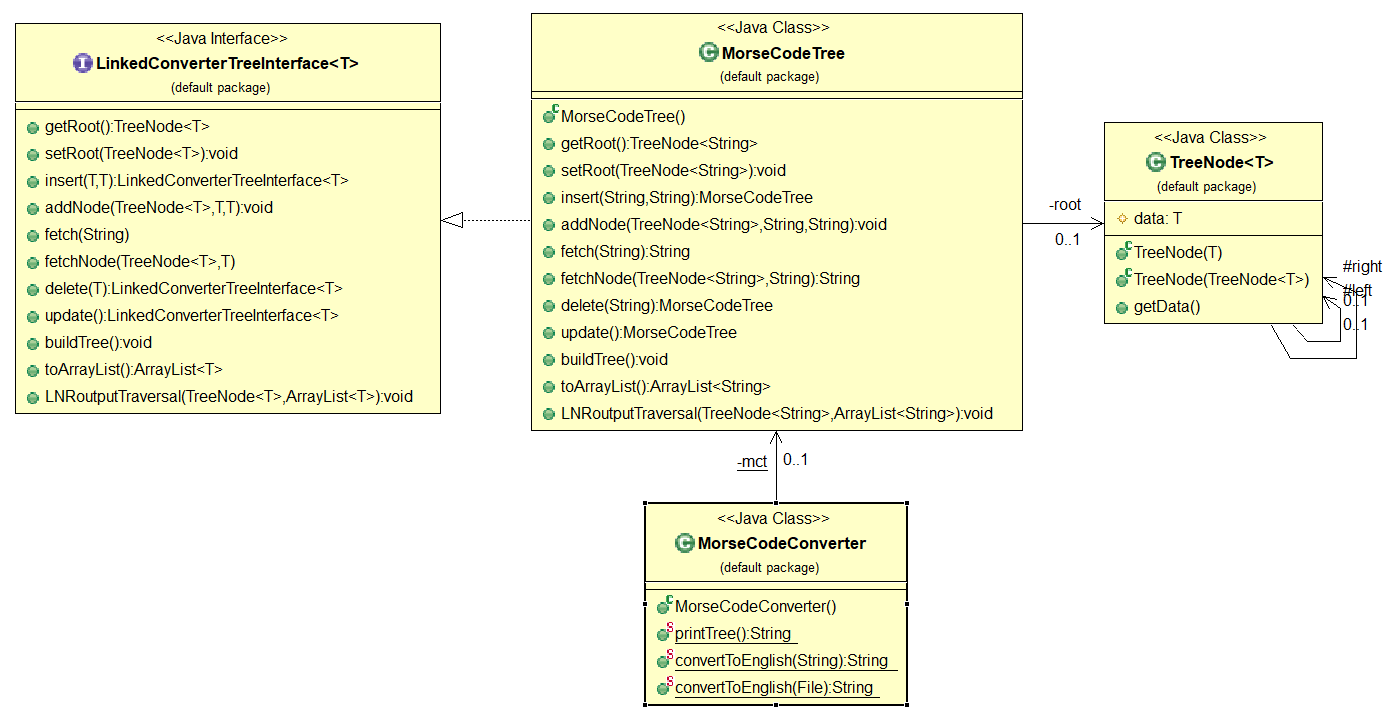
**MorseCodeConverter:**

* Make a static object of MorseCodeTree which calls the MorseCodeTree constructor
* Create a printTree method that returns a string with all the data in the tree in LNR order with a space in between them.
* Create two static methods convertToEnglish
  + Pass a string object (morse code) in one method and a file containing morse codes in the other
  + These static methods use the MorseCodeTree to convert from morse code to English characters.

**Test Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | .... .. / .. / .- -- / -.- .- -... .. -. -.. .-. .- | .... .. / .. / .- -- / -.- .- -... .. -. -.. .-. .- | Hi I am Kabindra | Hi I am Kabindra | Yes |
| 2 | .... . .-.. .-.. --- / .-- --- .-. .-.. -.. / -. .. -.-. . / - --- / -- . . - / -.-- --- ..- | .... . .-.. .-.. --- / .-- --- .-. .-.. -.. / -. .. -.-. . / - --- / -- . . - / -.-- --- ..- | Hello world nice to meet you | Hello world nice to meet you | Yes |
| 3 | -- --- -. - --. --- -- . .-. -.-- / -.-. --- .-.. .-.. . --. . | -- --- -. - --. --- -- . .-. -.-- / -.-. --- .-.. .-.. . --. . | Montgomery college | Montgomery college | Yes |

**UML Diagram**

****

**Learning Experience**

We were required to work with morse code in this assignment. We were required to write the classes required to create a Morse Code Converter Utility using a generic linked binary tree with generic TreeNodes to convert Morse Code into English. This assignment required the concepts of generic classes, utility class (all static methods), linked trees and building a tree for conversion purposes. We were also required to make out own Junit test classes.

Working with the morse codes was very interesting. I learned the implementation of trees and linked trees. The provided Javadoc was helpful as always. This project was fairly straight and easy as it did not have a lot of parts to it.

This code went fairly smooth compared to some previous assignments. Looking forward for the next project.