Student Name: Calvin Moylan

Student ID: 30018702

Date: 28/08/2020

Assessment Title: Portfolio One

## AT 2: Activity 2

## Design Specifications

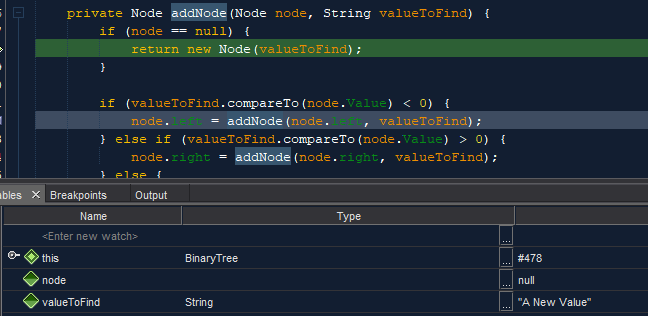
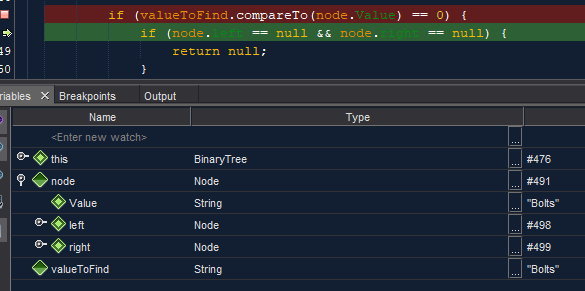
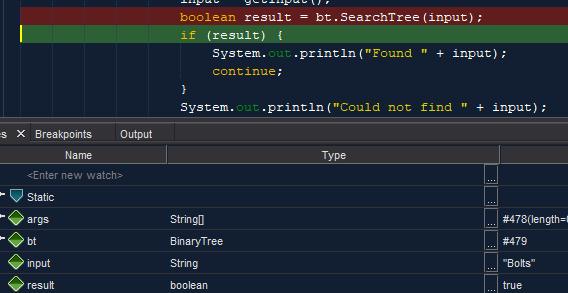
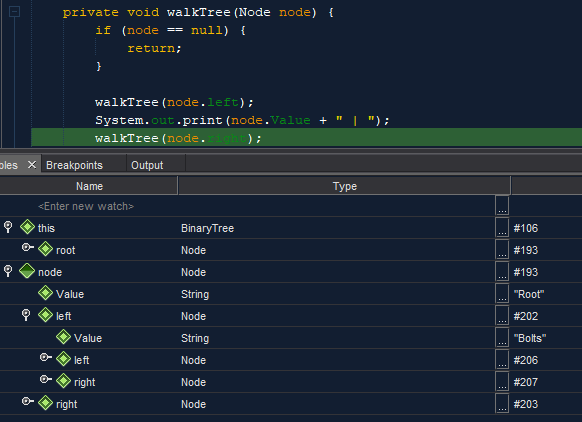
The application in question is a binary search tree. It will store a list of mechanical parts which will be sorted alphabetically The purpose of this application is to store a list of 10 or more parts in a binary tree which will have the ability to search, add and remove from the tree.

## Design Specifications Analysis

For this task, I have followed all requirements for the task, which includes searching, adding and deleting from a binary tree which stores 10 or more parts in it.

During production of the task I decided to make the end user experience a little better by having an infinite loop where they can type to select what they would like to do with the application, add, search, remove and walk the tree. I thought that was better than the user having to restart the program each time they were done with one option.

## Debugging

Adding a value:  
Removing a value:  
  
Searching for value:  
  
Walking Tree:  


## Testing the Application

### Test and validate with sample inputs with screenshots.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Test Case** | **Data** | **Expected Result** | **Actual Result / Comment** |
| *Case 1* | *Traversing a Binary Tree* | *Default Binary Tree Values* | *The values at all nodes are outputted to the console.* | *Results as expected.*  *Ref Figure 1.1* |
| *Case 2* | *Adding to binary tree* | *Default Binary Tree values + user input* | *The value is added to the binary tree* | *Results as expected.*  *Ref Figure 1.2 & 1.3* |
| *Case 3* | *Searching for a value that exists* | *Default Binary Tree values + user input* | *The searched value is found and user is told* | *Results as expected.*  *Ref Figure 1.4* |
| *Case 4* | *Searching for a value that doesn’t exist* | *Default Binary Tree values + user input* | *The value is not found and no output is given* | *Results as expected.*  *Ref Figure 1.5* |
| *Case 5* | *Deleting a part member from the binary tree* | *Default Binary Tree values + user input* | *The part member is removed from the binary tree* | *Results as expected.*  *Ref Figure 1.6* |
| *Case 6* | *Deleting a part member that doesn’t exist* | *Default Binary Tree values + user input* | *Value isn’t found and nothing happens to the binary tree* | *Results as expected.*  *Ref Figure 1.7* |



Figure 1.

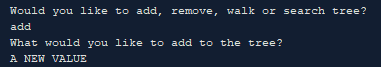


Figure 1.2

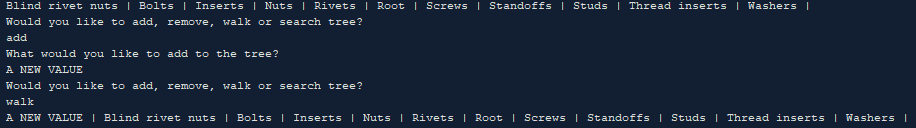


Figure 1.3

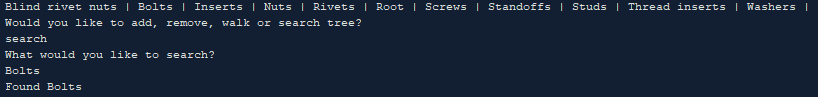


Figure 1.4

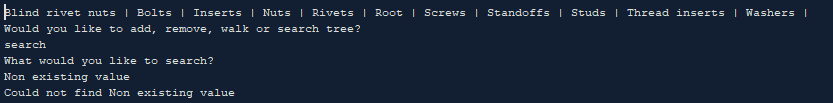


Figure 1.5

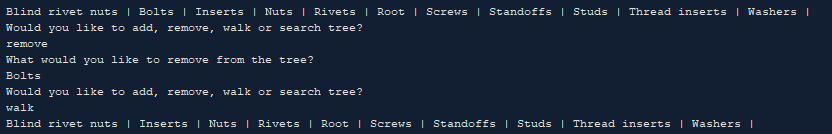


Figure 1.6

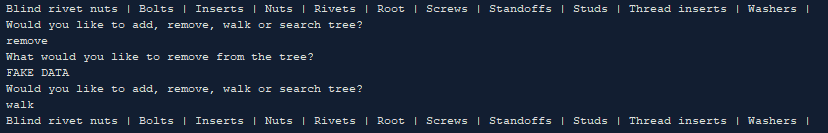


Figure 1.7

END OF TEST TABLE

## Implementation

The application has been uploaded to GitHub under the url: <https://github.com/CalvinMoylanTAFE/JavaThreePortfolio/tree/main/QuestionTwo>  
Using GitHub, we will be able to update the repository if any future changes are necessary such as a bug fix or a future feature. The application can be implemented on any computer that supports Java Runtime Environment so deploying the application will require JavaRE to be installed.