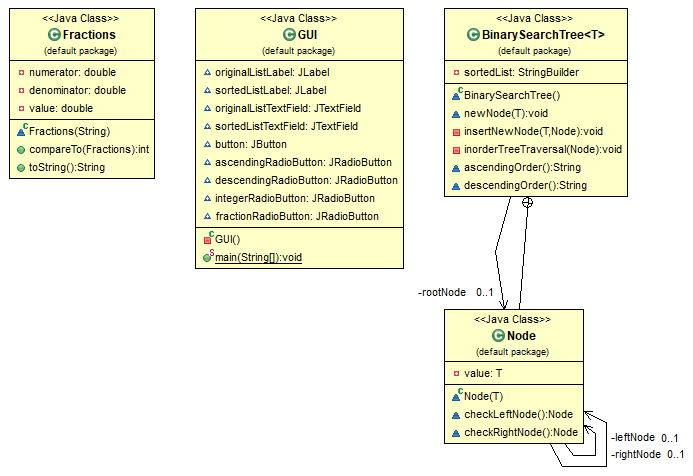
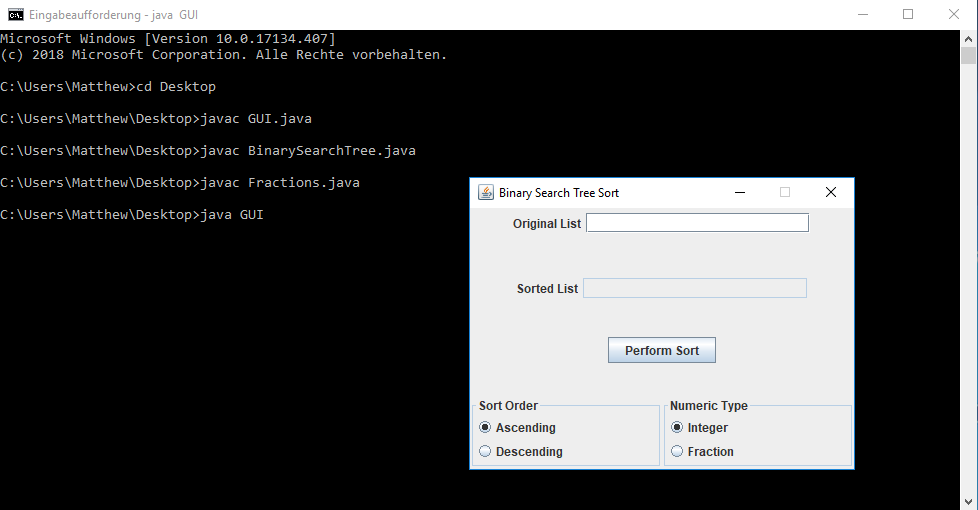
**UML Class Diagram**



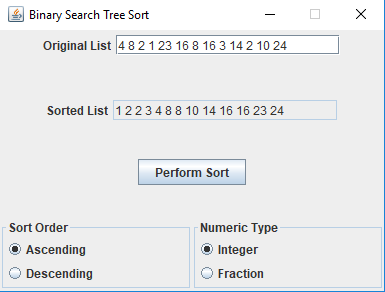
**Test Case Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected Output** | **Actual Output** | **Pass?** |
| **Test Case #1**  **4 8 2 1 23 16 8 16 3 14 2 10 24**  **Sort Order: Ascending**  **Numeric Type: Integer** | **1 2 2 3 4 8 8 10 14 16 16 23 24** | **1 2 2 3 4 8 8 10 14 16 16 23 24** | **Yes** |
| **Test Case #2**  **4 8 2 1 23 16 8 16 3 14 2 10 24**  **Sort Order: Descending**  **Numeric Type: Integer** | **24 23 16 16 14 10 8 8 4 3 2 2 1** | **24 23 16 16 14 10 8 8 4 3 2 2 1** | **Yes** |
| **Test Case #3**  **1/2 3/4 3/2 5/8 4/9 7/16 5/32 1/8**  **Sort Order: Descending**  **Numeric Type: Fraction** | **3/2 3/4 5/8 1/2 4/9 7/16 5/32 1/8** | **3/2 3/4 5/8 1/2 4/9 7/16 5/32 1/8** | **Yes** |
| **Test Case #4**  **1/2 3/4 3/2 5/8 4/9 7/16 5/32 1/8**  **Sort Order: Ascending**  **Numeric Type: Fraction** | **1/8 5/32 7/16 4/9 1/2 5/8 3/4 3/2** | **1/8 5/32 7/16 4/9 1/2 5/8 3/4 3/2** | **Yes** |
| **Test Case #5**  **6 2 8 13 6 4 5 x 27 3**  **Sort Order: Ascending**  **Numeric Type: Integer** | **For input string: “x”**  **Invalid Input.**  **Please enter a valid numeric expression.** | **For input string: “x”**  **Invalid Input.**  **Please enter a valid numeric expression.** | **Yes** |
| **Test Case #6**  **1/4 2/3 6/5 7/0 4/8 2/5 3/8 6/5**  **Sort Order: Descending**  **Numeric Type: Fraction** | **7/0**  **Invalid Input.**  **Please enter a valid numeric expression.** | **7/0**  **Invalid Input.**  **Please enter a valid numeric expression.** | **Yes** |
| **Test Case #7**  **2/1 4/5 3/4/8 7/8 3/2 5/6 4/7 7/6**  **Sort Order: Ascending**  **Numeric Type: Fraction** | **3/4/8**  **Invalid Input.**  **Please enter a valid numeric expression.** | **3/4/8**  **Invalid Input.**  **Please enter a valid numeric expression.** | **Yes** |

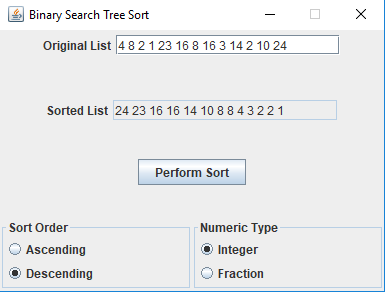
**Screen Capture of Successful Compilation & Execution of Java Code**



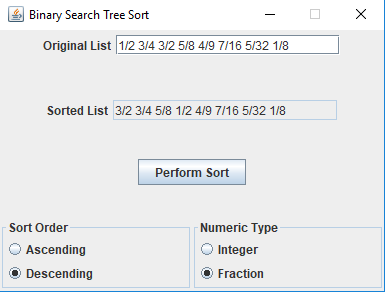
**Screen Capture of Test Case #1**



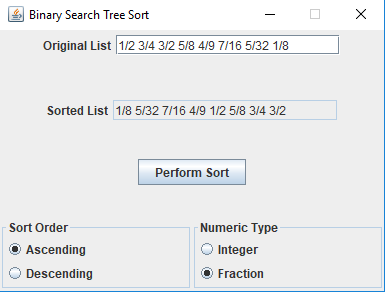
**Screen Capture of Test Case #2**



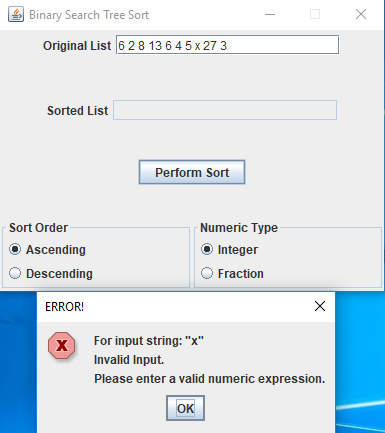
**Screen Capture of Test Case #3**



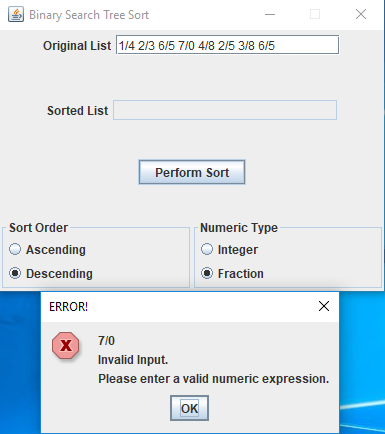
**Screen Capture of Test Case #4**



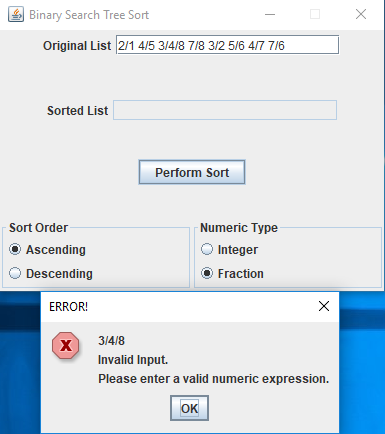
**Screen Capture of Test Case #5**



**Screen Capture of Test Case #6**



**Screen Capture of Test Case #7**



**Lessons Learned**

This was a fairly time consuming project, but I feel that I learned quite a bit from working on it. I feel the most import thing I learned was how to implement the *Comparable* interface with its accompanying *compareTo* method, to sort lists and arrays. In this project, it was used to sort fractions. I can see how this could come in handy when designing data structures. I foresee myself using this interface in the future. Furthermore, I got more experience implementing binary search trees. Overall, it was a fun project to work on.