Antonio Scalfaro

Project 3 – BST

CMSC 315

02/20/2024

This is the BST program which is designed to take input in from the user in a specific format and create a binary search tree given the input. The input must be in form (Root (left subtree) (right subtree)) with asterisks indicating null children. This program has three classes: BinaryTree, InvalidInputException, and Main. There is technically a fourth class in the program but is imbedded as a private class in the BinaryTree class called TreeNode, a class to create each node in the BinaryTree. The BinaryTree class is where the majority of the functionality comes from, it contains one private class (containing just a constructor for TreeNode) and 18 class methods. The class methods are all listed in the class UML diagram so I will forgo repeating the information here. The program is initiated by Main and produces a welcome message with program details and a loop menu for the user to continue to input a bst string which will produce a tree if in correct form or throw the custom InvalidInputExcpetion if incorrect. The user will be prompted for a new bst string while the user continues to input ‘y’ when prompted if they want to continue, regardless if the previous bst string produced an exception.

**Test Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **Input** | **Expected Ouput** | **Actual Output** | **Pass (Y/N)** |
|  | “n” | “Would you like to create a bst? y/n” “Thanks for using the bst creator program” | “Would you like to create a bst? y/n” “Thanks for using the bst creator program” | Y |
|  | “ y “, “(53 (28 (11 \* \*) (41 \* \*)) (83 (67 \* \*) \*))”, “n” | “Would you…”, “Enter bst”, “Your BST:  53  28  11  41  83  67  This is a binary search tree and is balanced  Height: 2  Add another Tree Y/N”, “Thanks for…” | “Would you…”, “Enter bst”, “Your BST:  53  28  11  41  83  67  This is a binary search tree and is balanced  Height: 2  Add another Tree Y/N”, “Thanks for…” | Y |
|  | “Y”, “(63 (51 (20 (13 \* \*) \*) \*) \*)”, “y”, “(53 (28 (11 \* \*) (41 \* \*)) (83 (67 \* \*) \*))”, “n” | “Would you…”, “Enter…”, “Your BST:  63  51  20  13  It is a binary search tree but not balanced  Original height: 3  20  13  51  63  Balanced height: 2  Add another Tree Y/N”, “Your BST:  53  28  11  41  83  67  This is a binary search tree and is balanced  Height: 2  Add another Tree Y/N”, “Thanks for…” | “Would you…”, “Enter..”, “Your BST:  63  51  20  13  It is a binary search tree but not balanced  Original height: 3  20  13  51  63  Balanced height: 2  Add another Tree Y/N”, “Your BST:  53  28  11  41  83  67  This is a binary search tree and is balanced  Height: 2  Add another Tree Y/N”, “Thanks for…” | Y |
|  | “y”, “exceptionCheck”, “y”, “(5 \* \*”, “y”, “5 \* \*)”, “y”, (5 \* \*)5”, “y”, “(5 \*)”, “n” | “Would you…”, “Enter..”, “Invalid preorder sting input: exceptionCheck”, “Add another…”, “Enter…”, “Invalid…”, “Add another …”, “Enter…”, “Invalid…”, “Add another …”, “Enter…”, “Invalid…”, “Add another …”, “Enter…”, “Invalid…”, “Add another …”, “Thanks for…” | “Would you…”, “Enter..”, “Invalid preorder sting input: exceptionCheck”, “Add another…”, “Enter…”, “Invalid…”, “Add another …”, “Enter…”, “Invalid…”, “Add another …”, “Enter…”, “Invalid…”, “Add another …”, “Enter…”, “Your BST: 5  This is a binary search tree and is balanced  Height: 0”, “Add another …”, “Thanks for…” | N – Fails last exception check “(5 \*)” |
|  | “y”, “(50 (25 \* \*))”, “n” | “Would you…”, “Enter…”, “Invalid…”, “Add another…”, “Thanks for … “ | “Would you…”, “Enter…”, “Invalid…”, “Add another…”, “Thanks for … “ | Y |
|  | “y”, “(13 (53 \* \*) (11 (59 \* \*) \*))”, “n” | “Would you… “, “Enter…”, “Your BST:  13  53  11  59  It is not a binary tree  Original height: 2  13  11  53  59  Balanced height: 2  Add another Tree Y/N”, “Thanks for…” | “Would you…”, “Enter…”, “Your BST:  13  53  11  59  It is not a binary tree  Original height: 2  13  53  59  11  Balanced height: 2  Add another Tree Y/N”, “Thanks for…” | N – Restructures tree in different structure than given in the pdf for project. |

**Test 1:**

A screenshot of a computer

Description automatically generated

**Test 2:**

A screenshot of a computer

Description automatically generated

**Test 3:**

A computer screen shot of a computer screen

Description automatically generated

**Test 4:**

A computer screen shot of a program

Description automatically generated

**Test 5:**

A screen shot of a computer

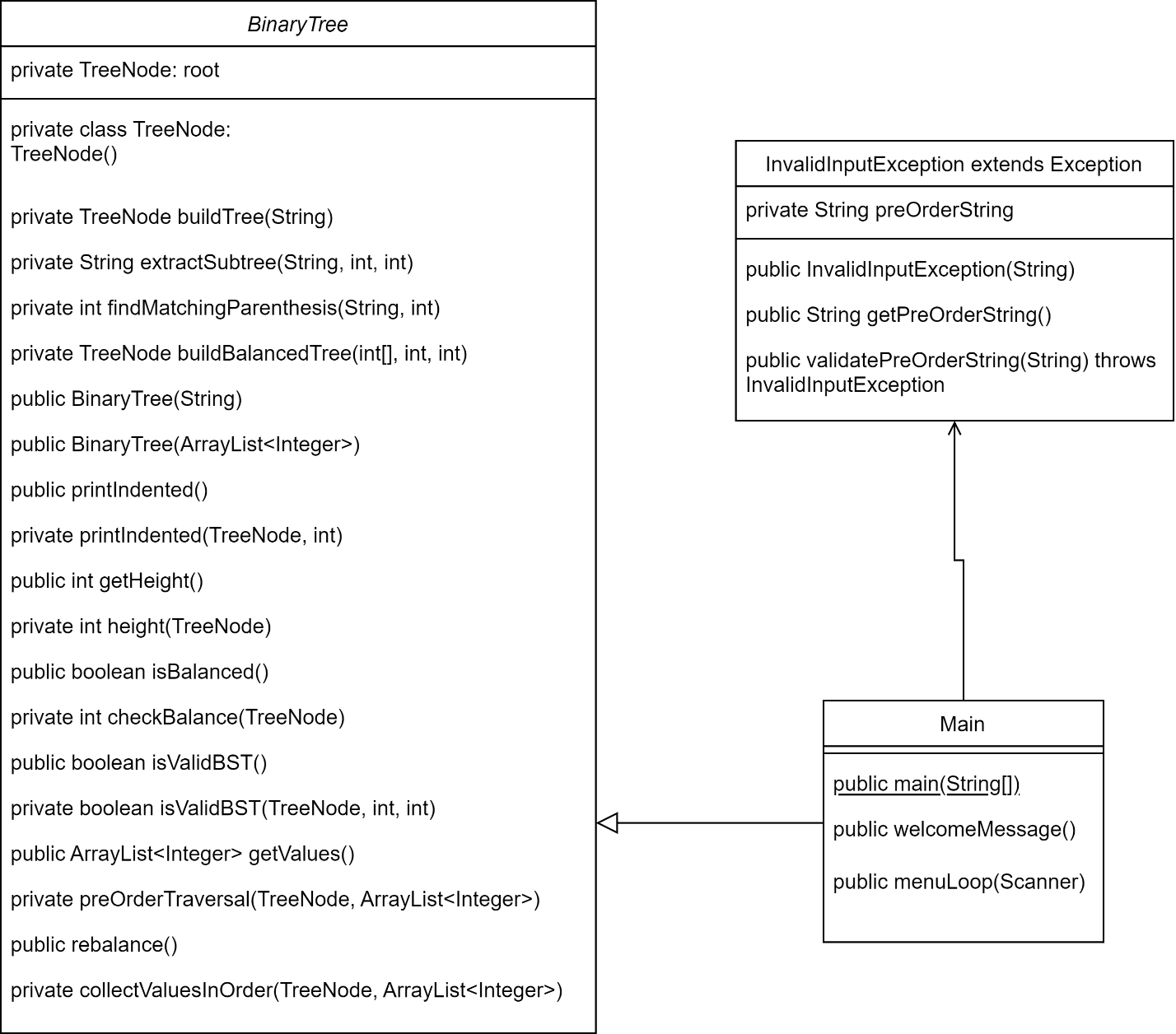
Description automatically generated

**Test 6:**

A computer screen shot of a computer screen

Description automatically generated

**Class Diagram:**

****

**What was Learned?**

This program was a very fun and challenging program to complete, emphasis on challenging! I found it to be very tough to get the logic quite right and I ended up implementing 18 total methods in the BinaryTree class which feels like quite a bit of methods to have in one class. With binary trees, recursion is so important and getting to use it more was good experience. I also utilized private ‘helper’ methods that utilized recursion so that the public accessible methods could still be called and produce the correct output (i.e. public int getHeight() calls private int getHeight(TreeNode)). This was the first project which I had test cases that failed upon testing, so there is some discussion and learning to be had there. I failed to catch an invalid input that comes in the form of a node having a single asterisk as a child (i.e (5 \*), should fail but produces a tree with root 5 and two null children). I wracked my brain for a while trying to figure it out but ultimately was stumped. Every change I made negatively effected performance with other input exceptions or in some cases correct inputs were thrown exceptions, I felt it better to leave it as is. The other test case that failed was the input (13 (53 \* \*) (11 (59 \* \*) \*)), in the pdf for the project it displayed a new bst that was in the form of (13 (11 \* \*) (53 ( 59 \* \*) \*)), however my program produces an output of (13 (53 \* \*) (59 (11 \* \*)\*)). As I stated earlier, this was a challenging project that really pushed me to think about the many possible input cases and in some respects I struggled. However, it gave me great perspective of where I am in my coding journey.