

Fall 2015
CSCI 2125: Assignment 2

Problem Description

For this homework, you need to download the following files from moodle:

- BinaryTree.java
- TestBinaryTree.java
- TestBinarySearchTree.java
- Startup.java

Part 1

The BinaryTree.java file contains an implementation of Binary Tree. However, signatures for the following two methods are provided but their bodies are not defined.

- width(): should compute and return the width of the tree.
- breadthFirstTraverse(): should traverse the tree in breadth-first order and return a string that represents the breadth-first traversal sequence of the tree.
- postOrderTraverse(): should traverse the tree in post order and return a string that represents the post order depth first traversal sequence of the tree.
- inOrderTraverse(): should traverse the tree in order and return a string that represents the in order depth first traversal sequence of the tree.

You need to implement the bodies of the aforementioned methods so that they perform the computation in accordance with what they are meant for. For this task you must use your own implementation of queue, that can grow in size when necessary. You must NOT make any other modifications in BinaryTree.java.

Currently the test code in TestBinaryTree.java fails due to empty implementation of the aforementioned two methods. Your implementation, if done correctly, should make all the tests pass. Do NOT modify or submit this provided test code.

Part 2

You need to write BinarySearchTree.java file that should include a generic implementation of a Binary Search Tree in BinarySearchTree class, which *must extend/inherit* from the provided Binary Tree class (defined in BinaryTree.java).

Your BinarySearchTree must provide the following methods, in addition to any other methods you want:

- public void insert(AnyType value): add the given element to the binary search tree (does nothing for duplicate entries).

- `public void remove(AnyType value)`: remove the given value from the binary search tree.
- `public boolean contains(AnyType value)`: returns true if the given value is found in the binary search tree, and false otherwise. Your implementation of `BinarySearchTree` must pass all the tests provided in `TestBinarySearchTree.java`. Your implementation must also work with the provided `Startup.java`. Do NOT modify or submit this provided test code.

Test Code

- Your implementation in `BinaryTree.java` must pass all the JUnit tests provided in `TestBinaryTree.java`.
- Your implementation of `BinarySearchTree` must pass all the JUnit tests provided in `TestBinarySearchTree.java`.

Your queue implementation must be accompanied by test code covering different corner cases. You need to *submit* your JUnit test along with your queue implementation.

For more about JUnit testing, please see the instructions available on moodle:

http://uno.mrooms3.net/pluginfile.php/1051326/mod_resource/content/1/JUnit_Testing.pdf

ReadMe.txt

When the description of the assignment does not explicitly specify something, you have the liberty to make your choices in those scenarios. Please mention any design choices you made including any reasoning behind your choices in a `ReadMe.txt` file. If you could not complete a certain part correctly, mention it in `ReadMe.txt`. If you did not find anything to include in the file, simply put your name and email address.

Startup.java

Provided `Startup.java` contains a main method, which can work as a starting point for your program. For this assignment, this `Startup` class acts as a sandbox for you to make use of your `BinarySearchTree`. Your implementation of `BinarySearchTree` must work with the provided `Startup.java`. Do NOT submit this provided `Startup.java`.

Clean Code

Make sure you have produced clean code as discussed in the first class. Please see the general guidelines for producing clean code available on moodle:

http://uno.mrooms3.net/pluginfile.php/1044879/mod_resource/content/1/CleanCodeInstructions.pdf

Deadline

November 6th 2015 midnight (Hard Deadline)

Submission

Keep all your files in a folder and zip it with the following format –

Firstname_Lastname.zip

Upload the zipped file in moodle before the deadline.