# Simulated coursework exercise - Binary Search Tree

An often used data structure in computer science is the binary tree structure. This is a graph-like structure where the values below each branch in the tree are sorted according to a comparison with the value at a particular node.

For example, one might have a tree where each value on the left of a node is less than the current value, and each value on the right is greater.

More information about binary search trees:

<https://en.wikipedia.org/wiki/Binary_search_tree>

Your task is to **implement a binary search tree** using the java language. You must use version **Java 8** or higher for this task.

Your solution should implement the following interfaces:

A class named "Tree", with methods:

* Tree(value) - constructs a new tree with this value as the root
* Smaller() – returns the tree that has smaller values than the current
* Greater() – returns the tree that has greater values than the current

If there is not a child in the expected location, simply return a null reference

* Val() – Returns the value of the tree
* Insert(value) – Inserts into the appropriate place in the tree the value. Returns true if it was inserted, false if it already existed

As long as the described interface is met, you may use any internal representation or extra methods as you like.

Some test files. These should follow the standard unit testing framework offered by **junit 4.12**. You should use these test files to check your own solutions, but remember that they will be used on other people's solutions:

* They should assume that the class "Tree" is in the same directory.
* They are only required to test what is offered by the interface above.

Within the coursework, you will also find some skeleton files that offer the interfaces, for you to fill in yourself.

Only submit source code files, not compiled classes, jar libraries or zipped IDE workspaces. You do not need to submit any libraries for use in junit, as the libraries required are already stored on the submission system.