

LAB 4 & HOMEWORK 4

February 7

Spring 2018

Abstract

In this lab we'll continue work on our functional binary tree design from class. To let us focus on trees we'll design a binary tree of **String** objects.

Begin by stubbing out the hierarchy containing a **BinTree** interface, **AbstractNode** abstract class, and **MTTree**, **Leaf**, and **TreeNode** concrete classes. For each class include:

- *private* default constructors and by-field constructors. Abstract classes should use protected constructors.
- public static factory methods for object construction¹
- Accessors (getters) for each field. No mutators (setters).
- **equals** and **hashCode**² methods
- **toString** (you may skip this for **AbstractNode**, provided you still print out all the necessary fields in each class)

The interface should initially include the following methods³:

- A method for computing tree size
- A method for computing tree height
- A method which fills an array⁴ of the Strings with the Strings contained in the tree in the order they'd be visited by an **inorder** traversal. Your design for this method should allow users to use a pre-allocated array and not require any extra arrays to be allocated in memory.

Lab time should be used for documenting, declaring, stubbing, and developing tests for the above listed methods. When that is complete or at the end of lab, use handin to submit your work as **lab4**. Implementation should then be completed for assignment **hwk4** which is due by the start of lab on **Wednesday 2/14**.

¹further discussion in lab

²Use Eclipse!

³see lecture notes 8 if you need a refresher on these

⁴<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>