TreeNode

-data: generic T
-left: TreeNode<T>
-right: TreeNode<T>

+getData(): T
+setData(T): void
+getRight(): TreeNode<T>
+getLeft(): TreeNode<T>
+setRight(TreeNode<T>): void
+setLeft(TreeNode<T>): void
+hasLeftChild(): boolean
+hasRightChild(): boolean
+equals(Object): boolean
+toString(): String

ExpTree

-root: TreeNode<String>
-infix: String

+getRoot(): TreeNode<String>
+getPostfixString(): String
+getInorderString(): String
+getInfixString(): String
+getPrefixString(): String
+equals(Object): boolean
+toString(): String
+getPostorderList(): List<TreeNode<String>>
+getPreorderList(): List<TreeNode<String>>
+getInorderList(): List<TreeNode<String>>
-isNumeric(String): boolean
-infixToPostfix(String): String
-priorityCheck(char): int
#pPostorder(TreeNode<String>): void
#pPreorder(TreeNode<String>): void
#pInorder(TreeNode<String>): void
#testEquality(TreeNode<String>, TreeNode<String>): boolean