

### Homework 3

DDL1: Nov 15, 23: 00

DDL2: Nov23, 23: 00

由于系里要求第十周不设置 DDL，以及很多同学有其他期中考试安排，故 DDL1 前提交得 100%，DDL2 前得 80%，DDL2 之后不得分

#### Binary Tree

(1)建立 BinaryTree 类，实现 AddLeft(),AddRight()等方法,通过添加节点的方式建立二叉树

(2)实现 inOrder, preOrder, postOrder, levelOrder 遍历，结果返回为 String

(3)根据 preOrder,inOrder 遍历构建树结构并输出 postOrder 遍历，根据 inOrder，postOrder 遍历构建树结构并输出 preOrder 遍历

```
public class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

    TreeNode(int x) {
        val = x;
        left = null;
        right = null;
    }
}

public class BinaryTree {
    public TreeNode root;

    public boolean AddLeft(TreeNode parent, TreeNode left){
    }

    public boolean AddRight(TreeNode parent, TreeNode right){
    }

    public String TraversalInOrder() {
    }

    public String TraversalPreOrder() {
    }

    public String TraversalPostOrder() {
    }

    public String TraversalLevelOrder() {
    }

    public String PreIn2Post(String preTravesal, String inTravesal){
    }

    public String InPost2Pre(String inTravesal, String postTravesal){
    }
}
```

## 测试程序

```
public class BinaryTreeTest {
    public static void main(String[] args) {
        BinaryTree binaryTree = new BinaryTree();
        TreeNode node0 = new TreeNode(3);
        binaryTree.root = node0;
        TreeNode node1 = new TreeNode(9);
        TreeNode node2 = new TreeNode(20);
        binaryTree.AddLeft(node, node1);
        binaryTree.AddRight(node, node2);
        TreeNode node3 = new TreeNode(15);
        TreeNode node4 = new TreeNode(7);
        binaryTree.AddLeft(node1, node3);
        binaryTree.AddRight(node1, node4);

        System.out.println("TraversalInOrder:" + binaryTree.TraversalInOrder());
        System.out.println("TraversalPreOrder:" + binaryTree.TraversalPreOrder());
        System.out.println("TraversalPostOrder:" + binaryTree.TraversalPostOrder());
        System.out.println("TraversalLevelOrder:" +
binaryTree.TraversalLevelOrder());

        System.out.println(binaryTree.PreIn2Post(binaryTree.TraversalPreOrder(),binaryT
ree.TraversalInOrder()));

        System.out.println(binaryTree.InPost2Pre(binaryTree.TraversalInOrder(),binaryTr
ee.TraversalPostOrder()));
    }
}
```

关于作业提交的一些要求:

- 不要 DDL 前几分钟才提交, 提交前检查压缩包内文件是否正确, 提交后把提交的文件下载后再检查一遍
- 没实现的方法也要写上方法名,并按要求的类型输出返回值,保证测试代码调用方法编译不出错
- 不要带 package
- 类名和方法名要符合要求
- 不要导入奇奇怪怪的包
- 不要用中文写代码, 注释也不要
- 如果代码在你本地都有红色警告就不要提交了, 不能编译的直接 0 分
- 什么都没做的不要交空文件,更不要交什么 word 文档
- 尽量不要使用 java8 以上的 java 新特性(能直接换过来环境就给分, 换不过来就没了)
- 把所有 java 文件打包[zip] 上传, 不要放进文件夹里, 也不要单独交 java 文件

