CSE 331 Section 5: Design Patterns and Nullness Checker

Design Patterns

1. Using what you know about design patterns, determine an appropriate design pattern for each of the following and explain:
   a) A Bank Account class, except all instance of bank accounts share the same balance.

   b) A MazeGame class with the capacity to support the creation of many different room types that extend the Room class.

   c) A NutritionFacts class that creates nutrition facts for food with different calories, fat, and sodium.

   d) A Point class, however creating duplicate Points should not result in higher memory usage.
Nullness Checker

2. Insert the correct type qualifiers (annotations) on the following code in order to prove it will not cause null pointer exceptions.

```java
public class BinarySearchTree {
    @__________
    private Node root;

    public BinaryTree() {
        this.root = null;
    }

    public void insert(@__________ String s) {
        this.root = insertHelper(i, root);
    }

    @__________
    private Node insertHelper(@__________ String s, @__________ Node root) {
        if (root == null) {
            return new Node(i, null, null);
        } else {
            if (root.data.compareTo(s) < 0) {
                root.left = insertHelper(i, root.left);
            } else {
                root.right = insertHelper(i, root.right);
            }
        }
    }

    private class Node {
        @__________
        public String data;

        @__________
        public Node left;

        @__________
        public Node right;

        public Node(@__________ String data, @__________ Node left, @__________ Node right) {
            this.data = data;
            this.left = left;
            this.right = right;
        }
    }
}
```