```
import java.util.concurrent.ConcurrentLinkedQueue;

{
	public static <T> String displayTree(TNode<T> t, int maxCharacters) {

		ConcurrentLinkedQueue<TNodeShadow> q = new
ConcurrentLinkedQueue<TNodeShadow>();
		String displayStr = "";
		int colWidth = maxCharacters + 1;
		int currLevel = 0, currCol = 0;
		TNodeShadow.columnValue = 0;

		if (t == null)
			return displayStr;

		TNodeShadow shadowRoot = buildShadowTree(t, 0);
		TNodeShadow currNode;
		q.offer(shadowRoot);

		while (!q.isEmpty()) {

			currNode = q.poll();
			if (currNode.level > currLevel) {
				currLevel = currNode.level;
				currCol = 0;
				displayStr += '\n';
			}

			if (currNode.left != null)
				q.offer(currNode.left);

			if (currNode.right != null)
				q.offer(currNode.right);

			if (currNode.column > currCol) {
				displayStr += formatChar(
						(currNode.column - currCol) * colWidth, ' ');
				currCol = currNode.column;
			}

			displayStr += formatString(colWidth, currNode.nodeValueStr);
			currCol++;
		}

		displayStr += '\n';
		shadowRoot = clearShadowTree(shadowRoot);
		return displayStr;
	}
```

```
private static <T> TNodeShadow buildShadowTree(TNode<T> t, int level) {

	TNodeShadow newNode = null;

	if (t != null) {

		newNode = new TNodeShadow();
		newNode.left = buildShadowTree(t.left, level + 1);
		newNode.nodeValueStr = t.value.toString();
		newNode.level = level;
		newNode.column = TNodeShadow.columnValue;
		TNodeShadow.columnValue++;
		newNode.right = buildShadowTree(t.right, level + 1);
	}

	return newNode;
}

private static TNodeShadow clearShadowTree(TNodeShadow t) {

	if (t != null) {
		t.left = clearShadowTree(t.left);
		t.right = clearShadowTree(t.right);
		return null;
	} else
		return null;
}

private static String formatString(int w, String s) {

	int sLength = s.length();
	if (sLength >= w)
		return s;
	String str = "";

	for (int i = 0; i < w - sLength; i++)
		str += " ";

	str += s;
	return str;
}

private static String formatChar(int w, char ch) {
	String str = "";

	for (int i = 0; i < w - 1; i++)
		str += " ";
```

```
            str += ch;

            return str;
        }

        private static class TNodeShadow {
            public static int columnValue;

            public String nodeValueStr; // formatted node value

            public int level, column;

            public TNodeShadow left, right;

            public TNodeShadow() {
            }
        }
}
```