Adding a "Null node" as a sentinel in binary trees. We can make the code for the methods we wrote in Days 10-11 much simpler by replacing every null pointer by a pointer to a special node, the NULLNODE. I have placed code for this new version in your repository, in the BinaryTreesWithNullNode project.

The NULLNODE is declared in the BinaryNode class:

```
public static final BinaryNode NULLNODE = new BinaryNode(null, null, null);
```

The second BinaryNode constructor becomes:

```
public BinaryNode(T elt) {
      this(elt, NULLNODE, NULLNODE);
   }
```

Example of simplification--the `contains` method:

	Original version	**With NULLNODE**

BinaryTree method	`public boolean contains(T obj){`
	`  return (this.root != null) &&`
	`  this.root.contains(obj);`
	`  }`
		`public boolean contains(T obj){`
		`  return this.root.contains(obj);`
		`}`

BinaryNode method	`public boolean contains(T obj) {`
	`  if (this.element.equals(obj))`
	`  return true;`
	`  return ((this.left != null) &&`
	`  this.left.contains(obj))`
	`  ||`
	`  ((this.right != null) &&`
	`  this.right.contains(obj));`
	`  }`
		`public boolean contains(T obj) {`
		`  return this != NULLNODE &&`
		`  (this.element.equals(obj) ||`
		`  this.left.contains(obj) ||`
		`  this.right.contains(obj));`
		`  }`

See the code in your repository for additional examples.