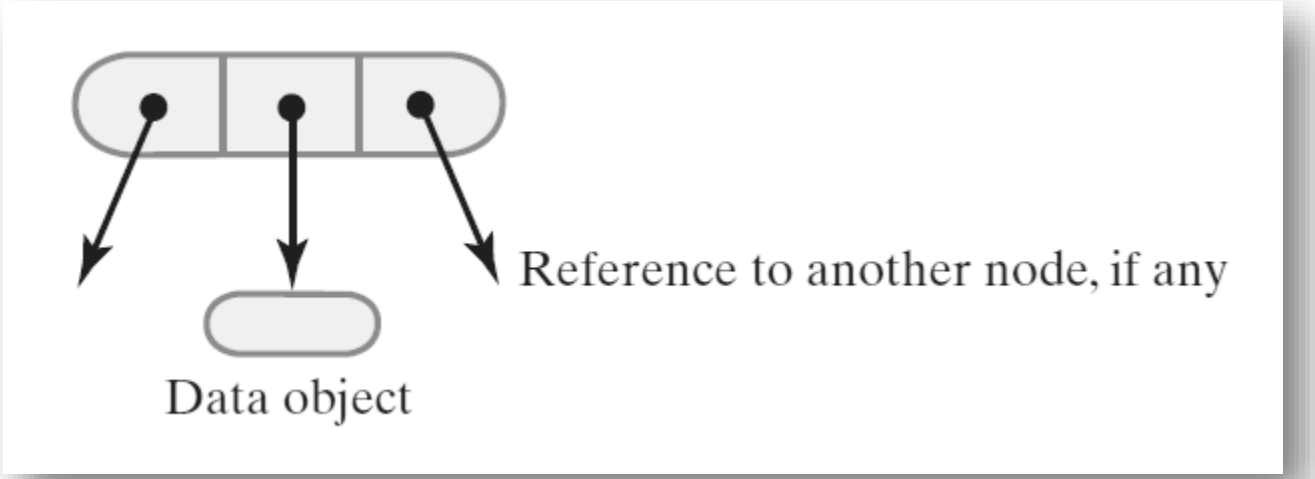
Tree Implementation

A Node in a Binary Tree



First Impl.

```
public class BinTree<E>
{
 private Node root;

 private class Node
 {
  public E data;
  public Node left;
  public Node right;

  public Node(E item)
  {
   if (item == null)
    throw new NullPointerException("Item 0");
   data = item;
   left = null;
   right = null;
  }
 }

 public BinTree()
 {
  root = null;
 }
```

```java
public BinTree(E item)
{
 if (item == null)
  throw new NullPointerException("Item 1");
 root = new Node(item);
}
```

```java
public BinTree(BinTree<E> lefttree, BinTree<E> righttree, E item)
{
 if (item == null) throw new NullPointerException("Item 2");
 if (lefttree == null) throw new NullPointerException("Ltree");
 if (righttree == null) throw new NullPointerException("Rtree");
 root = new Node(item);

 root.left = copyTree(lefttree.root);
 root.right = copyTree(righttree.root);
}
```

```java
private Node copyTree(Node r)
{
 if (r == null) return null;
 Node retval = new Node(r.data);
 retval.left = copyTree(r.left);
 retval.right = copyTree(r.right);
 return retval;
}
```

```java
public void preOrderTraversal()
{
 System.out.println("\nPre-order traversal");
 recPreOrderTraversal(root);
}
```

```java
private void recPreOrderTraversal(Node r)
{
 if (r == null) return;
 System.out.println(r.data.toString()); // "Visit the node"
 recPreOrderTraversal(r.left);
 recPreOrderTraversal(r.right);
}
```

```java
public void inOrderTraversal()
{
 System.out.println("\nIn-order traversal");
 recInOrderTraversal(root);
}
```

```java
private void recInOrderTraversal(Node r)
{
 if (r == null) return;
 recInOrderTraversal(r.left);
 System.out.println(r.data.toString()); // "Visit the node"
 recInOrderTraversal(r.right);
}
```

```java
public void postOrderTraversal()
{
 System.out.println("\nPost-order traversal");
 recPostOrderTraversal(root);
}
```

```java
private void recPostOrderTraversal(Node r)
{
 if (r == null)
  return;
 recPostOrderTraversal(r.left);
 recPostOrderTraversal(r.right);
 System.out.println(r.data.toString()); // "Visit the node"
}
```

Reference

- F. C. Carrano & T. M. Henry, “Data Structures and Abstractions with Java”, 4th ed., 2015. Pearson Education, Inc.
- BinTree.java