点类一样，是一个嵌套类。

```
private static class BinaryNode<AnyType>
{
        // Constructors
    BinaryNode( AnyType theElement )
      { this( theElement, null, null ); }

    BinaryNode( AnyType theElement, BinaryNode<AnyType> lt, BinaryNode<AnyType> rt )
      { element  = theElement; left = lt; right = rt; }

    AnyType element;            // The data in the node
    BinaryNode<AnyType> left;   // Left child
    BinaryNode<AnyType> right;  // Right child
}
```

图 4-16　BinaryNode 类

图 4-17 显示 BinarySearchTree 类架构，其中唯一的数据域是对根节点的引用，这个引用对于空树来说是 null。这些 public 方法使用了调用诸 private 递归方法的一般技巧。

现在描述某些私有方法。

```
public class BinarySearchTree<AnyType extends Comparable<? super AnyType>>
{
    private static class BinaryNode<AnyType>
      { /* Figure 4.16 */ }

    private BinaryNode<AnyType> root;

    public BinarySearchTree( )
      { root = null; }

    public void makeEmpty( )
      { root = null; }
    public boolean isEmpty( )
      { return root == null; }

    public boolean contains( AnyType x )
      { return contains( x, root ); }
    public AnyType findMin( )
      { if( isEmpty( ) ) throw new UnderflowException( );
        return findMin( root ).element;
      }
    public AnyType findMax( )
      { if( isEmpty( ) ) throw new UnderflowException( );
        return findMax( root ).element;
      }
    public void insert( AnyType x )
      { root = insert( x, root ); }
    public void remove( AnyType x )
```

图 4-17　二叉查找树架构