

자료구조 실습 보고서

[제11주] BST & AVL

2018/06/04

201402414 장수훈

```

21 public Object getRoot(){
22     return root;
23 }
24
25 public Object setRoot(Object root){
26     return this.root = root;
27 }
28
29 public BinaryTree getLeft()
30     return left;
31 }
32
33 public BinaryTree setLeft(BinaryTree left){
34     return this.left = left;
35 }
36
37 public BinaryTree getRight(){
38     return right;
39 }
40
41 public BinaryTree setRight(BinaryTree right){
42     return this.right = right;
43 }

```

<Binary Tree클래스-1>

[21~41] root, left, right에 대한 getter 와 setter를 생성하였다.

```

45 public Boolean isBST(){
46     if( !(root instanceof Comparable) ) return false;
47     Comparable key = (Comparable) root;
48     if( this.left != null){
49         if( key.compareTo(this.left.getRoot()) < 0) return false;
50         else return this.left.isBST();
51     }
52     if( this.right != null){
53         if( key.compareTo(this.right.getRight()) > 0) return false;
54         else return this.right.isBST();
55     }
56     return true;
57 }
58
59 public void insertKey(Object data){
60     if( ((String)root).compareTo((String)data) < 0){
61         if(this.right != null){
62             right.insertKey(data);
63         }
64         else{
65             this.right = new BinaryTree(data);
66         }
67     }
68     else if( ((String)root).compareTo((String)data) > 0){
69         if(this.left != null){
70             left.insertKey(data);
71         }
72         else
73             this.left = new BinaryTree(data);
74     }
75 }
76 }
77

```

<Binary Tree클래스-2>

[45~56] 이진탐색트리인지 확인 한 후 true 와 false를 반환하였다. 여기서 compareTo 함수를 사용하였는데 이의 형식은 key.compareTo(this.left.getRoot()); 일때 key 와 this.left.getRoot()를 비교하여 같으면 0을 반환하고 key가 크면 (앞에것이 크면) 양수, 반대일때는 음수를 반환하는 함수이다. 이진 탐색트리일 경우 true 아닐경우 false를 반환하였다.

[45~56] 이진탐색트리의 조건에 맞게 데이터를 삽입한다. Root와 삽입하는 데이터와 비교하여 삽입데이터가 큰경우 right가 null이 아닐때는 right로 돌아가 재귀를하고 아닐경우 데이터를 right에 삽입했다.

삽입하는 데이터가 root보다 작을 경우는 left로 재귀를 돌게하였다.

<결과화면 출력>

Console

```
<terminated> Main (6) [Java Application] C:\WF  
treeA: [[[A],B],C,[D,[[E,[F]],G]]]  
treeA.isBST(): true
```

실습 트리 그림

